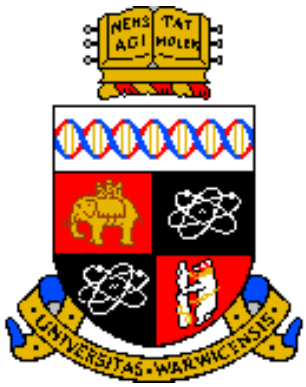


Dynamics of the Upper Airway

by

Tibor Sándor Bálint

A thesis submitted in partial fulfillment of the requirements
for the degree of

Doctor of Philosophy in Engineering

University of Warwick, School Engineering
Fluid Dynamics Research Centre

March 2001

Contents

List of Figures

List of Tables

Acknowledgements

I wish to thank my supervisors Dr. A.D. Lucey and Prof. P.W. Carpenter for they invaluable guidance throughout my research. Dr. Lucey's ongoing support and advice provided the environment, which allowed for a better understanding of the research topic and for a focused research without delays. Prof. Carpenter's feedback contributed greatly to my progress and his support on behalf of the department allowed me to practically dedicate the past two years to my studies on a full time basis.

I also wish to thank the University for allowing an extension during which time I was able to concentrate on the completion of my studies. Special thanks the University's IT Help and Advisory Desk for the computing resources and support.

The Navier-Stokes FEM formulation was partially based on course work and lecture notes of the postgraduate course at the University of Toronto, Canada. The course was entitled "MIE 1210S Computational Fluid Mechanics and Heat Transfer" and was offered during the Spring term of 1999. The lectures were given by Prof. M. Paraschivoiu, whose supervision during that course is greatly appreciated.

In addition I would like to thank my Aunts and my dear friends for their encouragement and support. And finally, I would like to dedicate this thesis to the memory of my Mother.

Declaration and inclusion of material from a prior thesis

I wish to declare that none of the material in this thesis was used before in previous research nor was it published.

The thesis is my own work and has not been submitted for a degree at another university.

This document is submitted to the School of Engineering, Civil and Mechanical Engineering Division, Fluid Dynamics Research Centre at the University of Warwick, for the degree of Doctor of Philosophy in Engineering.

All material in this document which is not my own has been identified.

Note on supplementary material

The thesis is submitted with a *Companion CD-ROM*, which includes the full thesis in *Adobe Acrobat (pdf)* format and an *MS-PowerPoint* presentation with additional material. These are: animations of the fluid flow and elastic wall deformation from the code validation phase and the results obtained using the coupled codes.

Further to this, all information presented in the thesis can be found on *CD-ROMs* submitted to the department, including the *FORTRAN* program source codes, executable files, scripts, figures, *pdf* and *postscript* files and supporting calculations.

Abstract

Fluid–structure interaction problems occur in everyday life, from medicine to aerospace engineering. One of the general questions from the field of bio-engineering is related to the interaction of a flexible cantilevered plate embedded in a channel flow. Further examining this problem, the motivation for the current study is the desire to elucidate the instability mechanisms of the upper airway with a view to understanding the medical conditions of snoring and obstructive sleep apnoea/hypopnea (OSAH). Snoring is caused by fluttering motions of the soft palate. OSAH involves complete or partial blockage of the airway due to a flow-induced collapse of the pharynx and/or large deflections of the soft palate thereby restricting the air flow to the lungs. The present work addresses the interaction of a flexible plate with viscous channel flow at Reynolds numbers up to 1500, which is about half of the typical Reynolds numbers for the *maximal* inspiratory flow rate of $0.001m^3/s$. To perform this investigation, a stable numerical simulation methodology for fluid–structure interaction was developed. To this end a computational model is constructed in which the fluid and flexible plate equations are solved concurrently. The Navier-Stokes equations are solved using an explicit finite-element method while the equations of motion of the plate are solved using an implicit finite-difference method; the latter permits plate deformations to evolve without prescription. A rapid mesh generator able to cope with an arbitrarily deforming geometry of the coupled problem has been implemented. Each of the fluid and solid computational elements of the method has been tested against a variety of unit and benchmark cases. The coupled model has then been used to perform a series of numerical simulations. For flow through both (oral and nasal) airways, the palate experiences a flutter-type of instability caused by a phase shift between the pressure difference across the plate and its motion. This yields an irreversible flow of energy into the plate. In contrast, when one airway is blocked, the plate loses its stability essentially due to divergence. In this case, a deformation is amplified because the pressure forces on the plate exceed the restorative structural forces in the plate; the amplification then serves to increase the force imbalance and further deflection growth ensues. Throughout the time series the full coupling method allows for unrestricted interaction between the viscous fluid flow and the flexible cantilevered plate. The computational model developed here holds much promise for the study of real, spatially varying, geometries and temporal variation of the mean flow. Moreover, the methodology can readily be extended to a similar configuration but with flexible channel walls more realistically representing the pharynx.

Chapter 1

Introduction

1.1 Overview

Studying the functions and dysfunctions of the upper airway, or blood flow in arteries belongs to the multidisciplinary field of bioengineering. It combines the knowledge base of engineering and medicine. Besides theoretical work, experimental studies and clinical measurements have advanced our understanding in the past two centuries. With the development of modern computers, computational fluid dynamics (CFD) was added, as an independent field, to theoretical and experimental studies of fluid motion. Numerical modeling of physical phenomena played an important role in the understanding of arterial flows as well[1].

While experiments can be expensive or measurement data hard to attain, properly validated computational simulations can hold the answer in characterizing conditions, such as flow and pressure, and can assist in the development of semi-empirical correlations.

Obstructive Sleep Apnoea / Hypopnoea (OSAH) is a common, yet relatively newly recognized condition. Understanding of the pathophysiology of OSAH has advanced rapidly in recent years. This condition produces major acute haemodynamic changes and is consequently linked

[1] A brief overview of blood flows in arteries is given in Appendix A, as an alternative application for the numerical method developed in the thesis. This line of numerical modeling is included only for illustration of the numerical method. Actual numerical simulations for such problems are not performed here. However, it is mentioned as a recommendation for future work.

to arterial hypertension and cardiovascular morbidity.

The role that the autonomic nervous system plays in mediating these cardiovascular changes has been the focus of intensive research activity and the development of new techniques in physiological monitoring. These techniques, such as spectral analysis of heart rate variability, blood pressure monitoring, measurement of muscle sympathetic nerve activity, radionuclide tests and animal models of OSAH have substantially increased our knowledge base.

Besides the observation and measurement of physiological and haemodynamic changes, computational fluid dynamic modeling can play an equally important role in understanding OSAH. Basic engineering fluid dynamic investigations can answer some of the questions related to the disorder and its dependence on contributing factors such as muscle tone and compliance.

This thesis documents a numerical investigation of flow–structure interaction in a two-dimensional channel with rigid and compliant sections, representing a simplified two-dimensional cross section of the upper airway.

The motivation for this analysis is a desire to gain better understanding of the respiratory system flow dynamics, specifically through the upper airway, under normal and abnormal physiological conditions.

To understand flow and pressure mechanisms, experimental – including clinical and laboratory measurements – and numerical studies are required under a wide range of conditions and configurations. Relevant experimental data is readily available in literature. While these experimental investigations are concerned with various aspects of upper airway flow conditions, it was found that there is no specific experimental data available to address the flow-structure interaction problem investigated in this thesis. Therefore, a systematic numerical examination

of phenomena outlined below is conducted. Numerical simulations include a simplified model of the upper airway with increasing geometric complexities, including compliant wall sections under steady flow conditions. The phenomena are examined by utilizing self-developed CFD codes. The codes solve the incompressible Navier-Stokes equations, using an explicit Finite Element Method (FEM) and the linear plate equations, using an implicit finite difference method. The flow solver is coupled with the wall solver for the compliant wall section.

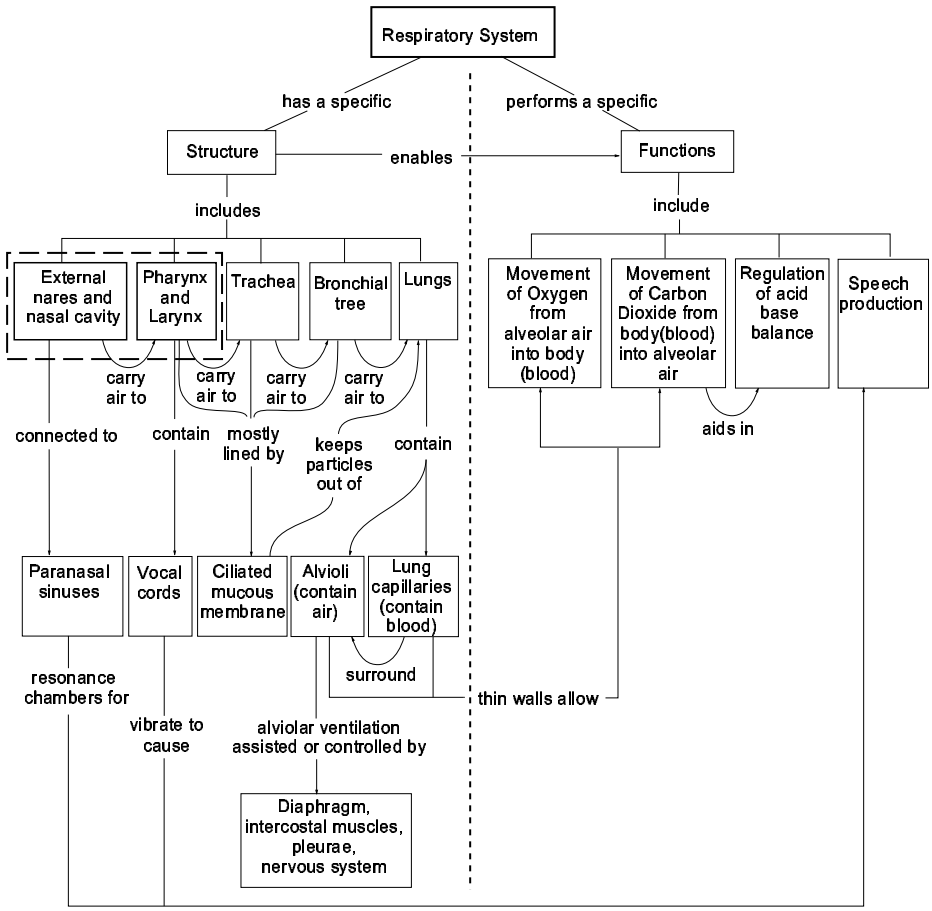
In the first stage, a prototype code was developed using MATLAB to solve the Navier–Stokes equations. In the second stage, the code is ported to FORTRAN 90/95 and further developed by modeling the compliant wall section. This improved code is used as production version during the numerical experiments. Runs are performed, validated and verified against various geometries and boundary conditions.

This introductory chapter consists of three additional sections. The second section introduces the upper airway, its place in the respiratory system, its anatomic structure and its functions and dysfunctions. It is relevant as an introduction to the motivating problems associated with this thesis. The third section introduces the planned validation and case matrix for this numerical investigation. The fourth section briefly outlines the software and hardware requirements for the research.

In summary, this introduction answers the following questions:

- What is the motivation for this thesis?
- Where is the upper airway and what is its role in the respiratory system?
- What are the functions and dysfunctions of the upper airway?
- What is OSAH?
- What validation and case matrix is planned for this study?

Figure 1.1: The respiratory system [SSF93]



- What are the software and hardware requirements to carry out these numerical investigations?

1.2 The upper airway

1.2.1 The respiratory system

This section gives a brief overview of the respiratory system and its functions. It also shows where the upper airway fits into the whole respiratory system, which is given as two boxes entitled “external nares and nasal cavity” and ”pharynx and larynx”. The respiratory system, its functions and the connection between them are demonstrated in Figure 1.1.

1.2.2 Anatomical structure of the upper airway

The upper airway with which we are concerned here extends from the nares and lips to the larynx. It is only a few centimeters long and a few square centimeters in cross section. It consists of a complex muscle structure, soft tissue and bones in its walls. It has many functions; it is a conduit for air, food and drinks. It also warms, humidifies and filters the air to the lungs. Cough reflex protects the lungs from inhaling foreign objects and clears secretions. Additional roles are gastrointestinal function and speech.

The major organs of the respiratory system are [otMHBL96] [Gra77]:

- the nose and mouth,
- the nasal and oral cavities,
- the pharynx (or throat),
- the larynx (or voice box),
- the trachea (or wind pipe),
- the bronci, bronchioles and lungs.

The incoming air enters the nose (and/or the mouth) and passes through the nasal cavity (and/or the oral cavity). The nasal cavity is the space inside the nose that lies between the floor of the cranium (or skull) and the roof of the mouth. It is divided into two halves by the septum (a dividing wall). Each half communicates with the outside via the nostrils and with the nasopharynx through the posterior (or internal) nares. (Internal nares are the openings leading from the nasal cavity into the pharynx.) The nasal and oral cavities are divided by the palate, also called the roof of the mouth. It consists of two portions: the hard palate and the soft palate. The hard palate, at the front of the mouth, is formed by processes of the maxillae (or upper jaw) and palatine bones and is covered by mucous membrane. The soft

palate, further back, is a movable fold of mucous membrane that tapers at the back of the mouth to form a fleshy hanging flap of tissue, called the uvula.

From the nasal and oral cavities the incoming air then continues through the pharynx (or throat). The pharynx is a 12-14 cm (about 5 inches) long tube, extending from the nasal cavity to the larynx (and esophagus). The pharynx is divided into four portions (see Figure 1.2), named for the structures behind which they lie, such as (1) *nasopharynx*, (2)*velopharynx*, (3)*oropharynx* and (4) *laryngopharynx*.

1. The *nasopharynx* has four openings and contains tonsils within its walls. Two openings are the internal nares, which are passageways into the nasal cavity and separated by the nasal septum. Two openings are passageways into the auditory – or Eustachian tubes – leading to the ears.

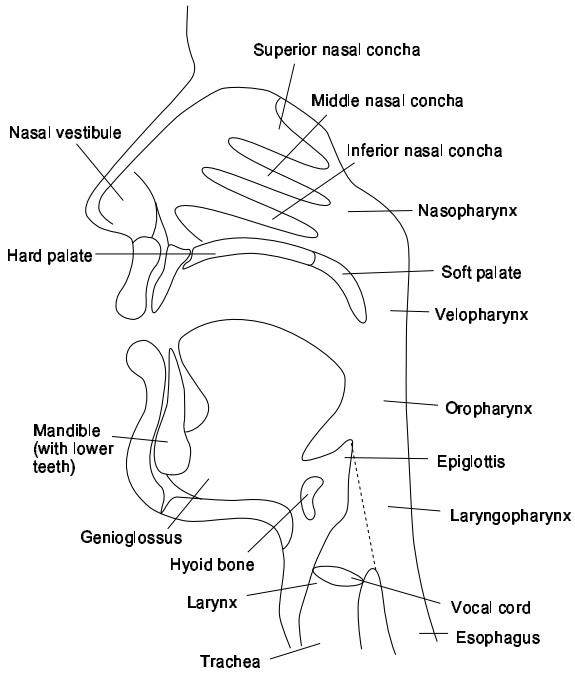
2. The *velopharynx* lies behind the soft palate between the nasopharynx and oropharynx.

3. *Oropharynx* is the part of the pharynx that lies between the soft palate and the hyoid bone. The hyoid bone is situated near the upper portion of the epiglottis and is a small isolated U-shaped bone in the neck. It supports the tongue and it is held in position by muscles and ligaments between it and the styloid process of the temporal bone. The epiglottis is a thin leaf-shaped flap of cartilage, covered with mucous membrane, situated immediately behind the roof of the tongue. It covers the entrance to the larynx during swallowing.

4. *Laryngopharynx* is the part of the pharynx that lies below the hyoid bone.

Vocal sound is produced in the *larynx*, within a pair of vocal cords. It also serves as an air passage conveying air from the pharynx to the lungs. It is situated in the front of the neck, above the trachea. It is made up of a framework of nine cartilages bound together by ligaments and muscles and lined with mucous membrane.

Figure 1.2: A midline segittal view of the upper airway



Parts of the upper airway are illustrated in Figure 1.2.

1.2.3 Airway resistance

There are several factors affecting upper airway resistance, such as:

- the affected anatomical sections of the upper airway (i.e., alae nasi, nasal cavity, pharynx, soft palate and larynx),
- gender, age, obesity and airway compliance,
- oral breathing (the oral cavity is also part of the upper airway),
- inspiration or expiration, and
- body position.

Alae nasy section: The anterior nose is a major but variable site of resistance [SOS82] and this is controlled by the *alae nasi* (the two lateral flared portions of the external nose) muscles.

Nasal cavity section: Further back in the *nasal cavity*, the two principal determinants of airway caliber are the skeletal structure of the nose and the thickness and vascular congestion of the nasal mucosa. Blood supply to the nasal mucosa is influenced by autonomical vascular tone. At any given time, one nostril has high resistance while the other has low resistance. This pattern alternates between the nostrils with a cycle of approximately 90 minutes.
Pharyngeal section: *Pharyngeal* resistance is determined by the structure of the facial bones, particularly the maxilla and mandible, and by pharyngeal lymphoid tissue, fat deposition in the pharyngeal walls and the activities of the upper airway muscles.

Obesity: is associated with decreased upper airway patency, principally related to increased fat deposition in the lateral walls of the pharynx.

Gender: is another important influence. Women tend to have smaller upper airways than men and lower pharyngeal resistance than men [BZH86].

Advancing age: is associated with reduction in upper airway size, with a redistribution of body fat from peripheral to truncal. This results in increased pharyngeal resistance in men and increase in airway compliance [BZH86].

The pharynx is the part of the upper airway, which is most vulnerable to collapse. Its surrounding *compliant walls* lack bony support and depend on the balance between the dilating and constricting muscle activities for maintenance of potency.

Velopharyngeal section and soft palate: The caliber of airway around the mobile *soft palate* depends on the position of the soft palate, which is controlled by the palatal muscles. These palatal muscles play an important role in determining the breathing route – oral or nasal –, although oral breathing also requires the mouth to be open. In general, quiet breathing at

rest is via the nose. When the breathing flow rate increases above 35-45 L/min – for example during exercise – ventilation is both nasal and oral.

The upper airway resistance is markedly reduced through *oral breathing*, however other functions, such as heating, humidifying and filtering of the inspired air is less efficient.

Laryngeal section: The *larynx* also offers some resistance that is greater in quiet expiration when the vocal cords adduct than in inspiration when they abduct. Thus the laryngeal muscles are the major determinants of laryngeal resistance. The larynx provides 25-30% of total airway resistance [TBLE91], with the glottic aperture between the vocal cords being the major site of laryngeal resistance.

Inspiration and expiration: The upper airway offers lesser resistance to airflow during inspiration than expiration [WLCZ85]. During *inspiration* the negative pressure tends to collapse the airway. This collapse can be counteracted by coordinated constriction of the upper airway muscles. During *expiration* the pharyngeal constrictor muscles contract, consequently breaking airflow, thus increasing airflow resistance. In normal adults this pharyngeal resistance is almost negligible [SH71].

Body position: The *position* of the head and neck also affects upper airway resistance. During either quiet breathing or panting, hypertension or hyperflexion of the head, resistance increases in the mouth and the larynx. The upper airway resistance is less while standing, compared to when lying horizontally, due to the gravity shifting the lower jaw (or mandible) and hypopharyngeal structures backward and to increase blood flow to the nasal mucous membrane (or mucousa). The upper airway is also less collapsible when sitting up at 30° compared with lying down.

1.2.4 Respiratory functions and dysfunctions of the upper airway

Besides providing passageway for air to the lungs, the upper airway also heats, humidifies and filters the air. In addition, it is involved in speech, swallowing and coughing. The complex muscle structure of the upper airway is vulnerable to functional problems that may compromise respiration. A collapsed upper airway increases upper airway resistance. This condition can occur even in people not suffering from upper airway dysfunctions. One important dysfunction is Obstructive Sleep Apnoea / Hypopnoea (OSAH) syndrome, where the collapse of the upper airway is so great that it affects respiration during sleep to an extent where arousal from sleep is required to restore the normal respiratory cycle. In consequence, this results in a number of physiological conditions, such as sleeplessness, hypoxia, nouropsychological and cardiorespiratory morbidity. Functional abnormalities of the larynx can also occur, including prolonged inspiratory laryngeal dysfunction or factitious asthma [PW99].

Sleep and wake cycles

The muscles of the upper airway are under both voluntary and involuntary control. While awake, the route of breathing can be determined consciously. During sleep most people breath through the mouth, although some breath through both nose and mouth. This means that the function of the upper airway muscles is affected by sleep, which abolishes voluntary control, but can also compromise involuntary or reflex control of the muscles. As a result, upper airway function is vulnerable during sleep and respiration is predisposed to dysfunctions in that state.

In normal subjects, the change in state from wakefulness to *Non-Rapid Eye Movement* (NREM) sleep induces a fall in ventilation and a rise in upper airway resistance. These changes occur within a breath of falling asleep and are the result of an abrupt reduction in activity in the diaphragm and in upper airway muscles.

During wakefulness, there is an immediate compensatory increase in pump muscle activity

when inspiratory resistance is increased, However, this compensatory response is delayed in sleep, with some partial recovery after several minutes. Tonic upper airway muscles, such as tensor palatine and the masseters, do not respond to increased resistance during sleep. Sleep also results in the marked reduction of the reflex pharyngeal dilator muscle response to sub-atmospheric upper airway pressure. In *Rapid Eye Movement* (REM) sleep, there are also reduced ventilatory and arousal responses to hypoxia, hypercapnia and upper airway irritation. Part of the rise in upper airway resistance in sleep can be explained by a fall in *Functional Residual Capacity* (FRC) of approximately 200 mL, although eliminating the fall in FRC only eliminates one-fifth of the increase of resistance.

In summary, sleep onset results in an immediate reduction of respiratory pump and upper airway muscle activity and reduction in ventilatory responses to chemical and mechanical stimuli. Consequently, ventilation is lower and upper airway resistance is higher in sleep than in wakefulness.

Obstructive sleep apnoea / hypopnoea syndrome

In Obstructive Sleep Apnoea / Hypopnoea (OSAH) syndrome the pharynx in the retropalatal and/or retroglossal area collapse. This can result in the *flutter* of the soft tissue walls of the upper airway, which manifests in snoring and/or in reduced ventilation producing a hypopnoea or, if the collapse is complete, an apnoea. With the increased resistance and respiratory effort during the apnoea, arousal from sleep occurs. This activates the upper airway muscles, opening up the airway and allowing resumption of ventilation. Sleep resumes after several breaths, then the upper airway collapses again and the cycle repeats. Associated with an apnoea are a fall in oxygen saturation and a rise in carbon-dioxide pressure. There is initial slowing of the heart rate (or bradycardia), followed by an increase in heart rate coincident with arousal and apnoea termination. Blood pressure rise during the apnoea and peaks following arousal and apnoea termination.

Symptomatic OSAH syndrome is present in 4% of men and in 2% of women. It is associated with hypertension and leads to daytime sleepiness, impotence, nocturia and increased motor vehicle accidents.

Pathogenesis of OSAH

There are *two major theories* about the pathogenesis of snoring and OSAH syndrome. In the *first*, the primary cause is either (1) an anatomically small airway [WM93] or (2) an excessively compliant upper airway [RL96].

1. Fat deposition in the pharyngeal wall – resulting in truncal obesity –, a large tongue – as in acromegaly, where the size of the head increases due to excessive production of growth hormones –, bony changes and enlarged lymphoid tissue in children may all reduce upper airway size.

2. In OSAH syndrome, the upper airway is typically compressed laterally so that it adopts an elliptical shape with the long axis in the anterior-posterior direction, whereas normally it is elliptical in the transverse plane.

   In OSAH patients, during inspiration, pressure becomes more sub-atmospheric than normal across the narrow pharynx, repeatedly activating the upper airway muscles while awake. However, in sleep, this reflex is diminished so that the dilator muscles are no longer active enough to prevent upper airway collapse. Consequently, an apnoea occurs.

In the *second* theory, the primary abnormality is deficient upper airway muscle function in sleep. This may be caused by narrowing or an altered shape of the upper airway, affecting the coordinated action of the dilator muscles and maintaining muscle tone. Those with snoring and sleep apnoea may have an exaggerated loss of upper airway muscle tone at sleep onset. As sleep progresses the activation of the upper airway muscles is delayed during inspiration. Simultaneously, the diaphragm applies a negative pressure to the upper airway before the airway dilator muscles become active.

Subjects with OSAH may also have central respiratory control instabilities leading to upper airway collapse and reduced upper airway response to chemical and mechanical stimuli in sleep. Muscle damage may also be a factor because the upper airway muscles are acting eccentrically and/or excessively to maintain muscle tone, leading to damage and/or hypertrophy, consequently affecting their function.

It is of course conceivable that the two mechanisms outlined in the two theories may coexist in any one patient or that different mechanisms may be responsible for producing the upper airway obstruction in different patients.

In addition, alcohol exaggerates the loss of upper airway muscle tone at sleep onset and this increases the tendency to obstruction in patients with OSAH syndrome. It also diminishes arousal responses to upper airway blockage, tending to worsen the degree of hypoxia occurring during sleep in OSAH patients. Women with reduced progesterone levels show higher occurrence of OSAH, although much of the OSAH syndrome in postmenopausal women can be explained by increased truncal obesity.

Management of OSAH

The standard treatment of OSAH is continuous positive airway pressure (CPAP). Positive pressure is applied through the nose via a mask. This prevents the collapse of the upper airway during sleep. CPAP eliminates OSAH in most patients. In this treatment a mask must be used every night for a long term, which makes it uncomfortable and difficult to use. Another treatment is tracheostomy. It is a surgical procedure, where a hole is made into the trachea through the neck. It is an inconvenient procedure to the patients. Some of the treatments target anatomical causes. These are upper airway surgery, mandibular advancement splinting, upper airway laser treatment and rapid maxillary (upper jaw) expansion. Surgically treating upper airway anatomy is only partially effective, which points to non-anatomical reasons as well. (Further information on the clinical manifestation and treatment of OSAH are given in recent

reviews [Whi95] [PN92].) It seems that OSAH syndrome is a functional problem. Therefore, new and effective treatments are needed to solve such a problem. Numerical investigations can contribute to our understanding of the physiological conditions associated with OSAH syndrome. Using relevant geometries and boundary conditions a properly verified and validated computer code can be employed to further understand the interaction between the flow and the upper airway walls, and to gain insight to flow instability, flutter and divergence of compliant surfaces.

1.2.5 Summary on the upper airway

The functions and dysfunctions of the upper airway are summarized below:

- The upper airway provides a passage for air to be breathed in an out of the lungs, it also heats, humidifies and filters the air and is involved in cough, swallowing and speech.
- The complex muscle structure of the upper airway that produces speech and swallowing in humans also modulates respiratory airflow throughout the respiratory cycle, but is vulnerable to functional problems that may compromise respiration.
- Even in normal people, there is some collapse of the upper airway and increased upper airway resistance during sleep.
- A substantial proportion of people suffer from Obstructive Sleep Apnoea / Hypopnoea (OSAH), in which the collapse of the upper airway is so great that respiration is compromised to the extent that arousal from sleep is required to restore adequate ventilation. The resulting disturbed sleep and hypoxia produce daytime sleepiness and nouropsychological and cardiorespiratory morbidity.
- Functional abnormalities of the larynx can also occur, including prolonged inspiratory laryngeal dysfunction or factitious asthma.
- There are several theories addressing the pathogenesis of Obstructive Sleep Apnoea. However, these theories can not fully explain OSAH.

- Therefore, further investigation is required to understand the mechanisms and conditions relevant to OSAH.

This thesis documents an investigation of the fluid dynamics conditions of the upper airway using computational methods with simplified geometries. The computational model, including the methodology and assumptions, are further explained in later chapters.

1.3 Validation and case matrix

This section introduces the validation and case matrix for testing the fluid flow, the moving wall and the coupled fluid-structure interaction cases. The basic concepts and definitions of code verification and validation used here are based on the document entitled *Guide for the Verification and Validation of Computer Fluid Dynamics Simulations* [AIA98]. For a brief overview refer to Appendix C.

Anytime one develops a numerical algorithm one would like to implement it in the form of a working code so that the algorithm's accuracy and efficiency can be demonstrated through computational experiments. In order to generate numbers, one must ultimately choose a particular physical problem for which the code provides numerical simulations. There are a wide variety of such test problems being used. The following section outlines the validation cases employed in this thesis. It includes the triangular and rectangular lid-driven cavity flows, backward and forward facing steps, flow around a vertical wall (fence) and a channel with split inlets and rigid or flexible wall sections. In addition, flexible plate oscillations are tested, based on analytical eigenmode solutions.

None of the problems discussed here are solvable exactly and therefore one can ask how one should measure the quality of the numerical solution. Ideally, and where possible, one should compare with experimental data. However, this is not a foolproof measuring device since one seldom knows the accuracy of the experimental data itself. One popular method of determin-

ing the accuracy of a numerical solution is to use the "eyeball norm" [Gun89]. A numerical solution is good in the eyeball norm if one looks at a picture of the numerical flow field and concludes that it looks reasonable, e.g., there are no unexpected wiggles. Of course, this measure of the quality of a flow simulation, although popular, can be extremely misleading. For example, one can easily generate numerical flow fields whose gross features, e.g., re-circulation regions, look reasonable but whose detailed features are wrong. A better way to judge the quality of the numerical solution is to compute using meshes of different sizes and show, using the type of norms that the convergence is apparent. Incidentally, the non-availability of exact solutions for most test problems points out the value of rigorous error estimates. After all, if we know *a priori* that the numerical solution converges to the (unknown) exact solution, and also know something about the rates of convergence, then we can have some confidence that we are producing meaningful numerical flow fields.

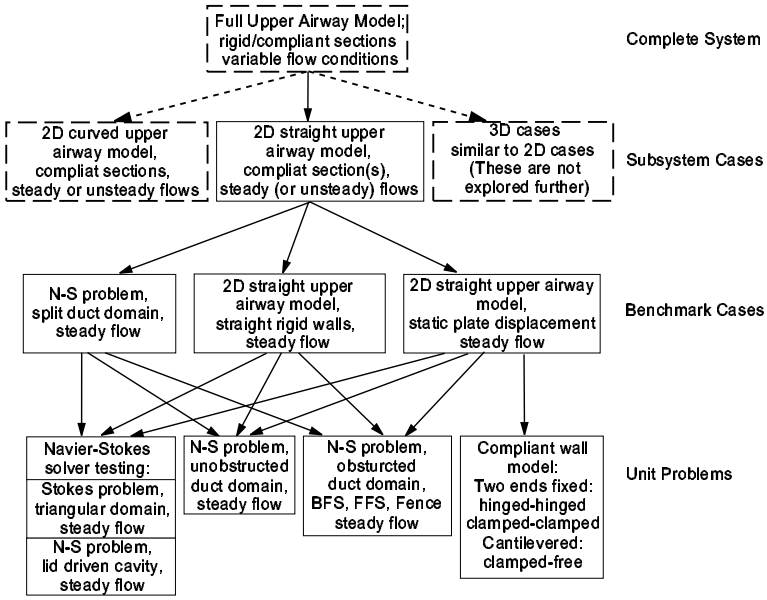
As shown in Figure 1.3, the simulations are performed for a partial validation matrix, up to the subsystem level. Boxes with continuous borders represent the test cases, while the boxes with dashed borders are included to show the validation phases - these phases are explained in the Appendix and shown in Figure C.3.

To be able to perform the numerical simulations, preliminary work is done, which for the fluid code included the following steps:

- Creation of a fast mesh generator for various domain configurations;
- Assembly of the elemental and global matrices and description of the block formation process;
- Selection of Δt from the Courant condition (for stability calculations) (CFL number).

Following this, the first level of the validation exercise contains the unit problems. These unit problems, which would test the FEM Navier–Stokes solver, are:

Figure 1.3: Case and validation matrix



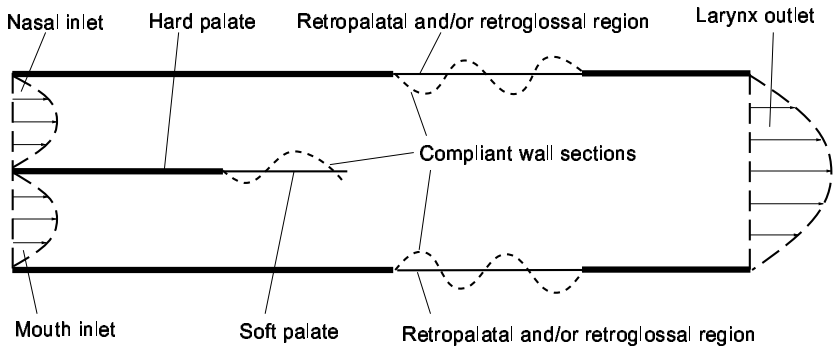
- Lid-driven cavity flow for triangular and square domains
- Poiseuille type duct flow
- Backward and forward facing steps
- Vertical wall obstruction in a duct

Unit problems for the flexible plate include:

- The first 6 eigenmodes of a flexible plate, fixed at both ends, using hinged–hinged boundary conditions;
- The first 6 eigenmodes of a flexible plate, fixed at both ends, using clamped–clamped boundary conditions;
- The first 6 eigenmodes of a cantilevered flexible plate.

The next step is to cover the more complex benchmark cases, where two elements of complex

Figure 1.4: Simplified model of upper airway



flow physics or relevant flow features are used. The three groups of cases for benchmarking are:

- Simplified two dimensional upper airway model, with partially split domain. The walls are rigid, and the flow steady. Either both inlets are open, or one of the inlets is blocked.
- Two dimensional model of the upper airway, with rigid walls, split inlet, with both inlets open, or one of the inlets is blocked.

The subsystem cases combine the models developed in the previous validation exercise steps, i.e., the coupled cases for the fluid–structure interaction. The geometry is still defined as a simplified two-dimensional upper airway model, with partially split domain. Either the compliant soft palate or both the compliant soft palate and the compliant pharynx are modeled, and the flow is either steady or unsteady. (See Figure 1.4). The final model also includes the proper variation in cross-sectional diameter changes. Note that the present analysis is limited to the modeling of the compliant cantilevered section only, using a uniform and steady inlet flow. From the simulations insight into flow stability and the flow–structure interaction are gained.

1.4 Software and hardware requirements

The runs were performed on a dual Pentium II (333 MHz) and on Pentium III (800 MHz) computers with 512 MB of memory. The computer codes were developed with Compaq Visual FORTAN 6.1 Professional version using FORTRAN 90/95 and various IMSL library functions. Additional coding and supporting scripts for code verification were developed using MATLAB 5.3, Student Edition. Symbolic manipulation was performed with MAPLE V Release 5 and MATLAB's Symbolic Toolbox. The document was typeset with LaTeX. Further documentation support was given using Tecplot 7 for data visualization, MS-Excel 97 for data processing and MS-PowerPoint for the presentation included on the *Companion CD-ROM*.

Chapter 2

Literature review

2.1 Fluid structure interaction

Fluid–structure interaction (FSI) plays an important role in predicting the effect of a flow field upon a structure and vice-versa. In general, we could categorize this interaction into three groups. In the first group, the flow field moves around stationary bodies, without any interaction between the two (e.g., wind loaded buildings). In the second group the wall boundaries move, but there is no feedback mechanism between the fluid and the structure. Airplane wings, turbine blades or guitar strings would belong to this group. The present study concentrates on the third group where the fluid–structure interaction refers to the interaction between a flow field and compliant or elastic structures. Here a mechanism is present to transfer energy between the flow field and the structure, influencing the stability of the flow field and/or the compliant structure. Laminar flow stability over compliant surfaces and channels can be influenced by the walls. The resulting instabilities are qualitatively different from those over rigid boundaries or channel walls. Besides the inertial and viscous forces (Re) influencing rigid surface instability, for flexible surfaces there are three additional properties, such as the elastic forces and material viscosity together with the geometry of the wall. Besides the Reynolds-number related forces this indicates additional parameters to induce or delay instability. For a flexible wall a normal velocity is allowed at the surface. Coupling between the mean flow and the surface can result in energy transfer between the flow and the wall. This could either

stabilize or destabilize the flow. An overview of instabilities over flexible surfaces are given by Kumaran [Kum00], describing these destabilizing mechanisms for internal flow. Kumaran classified the instabilities in a flow past flexible surfaces into three types, based on the asymptotic regime in which they were observed, the flow structure and the scaling of the critical Reynolds-number with the ratio of the elastic and viscous forces and the mechanism that destabilize the flow. His analysis was performed using numerical models of flow over a gel bed or in a tube with gel walls.

FSI cases can also be classified into one of two categories based on their geometric configurations. Unbounded flows refer to cases where developing boundary-layers over surfaces do no connect. In contrast, for bounded flow cases, the boundary-layers connect and the internal flow fields are affected.

The different kinds of phenomena that can be modeled using FSI techniques include a wide range of applications, from a flag flapping in the wind [ZCLS00] [Hub00] (for unbounded flows) to important applications in bio-engineering, such as heart-valve designs or modeling of the upper airway or blood flow in arteries (for bounded flows). Before progressing to these internal flow examples a brief summary of the unbounded flows is given. It is worthwhile examining the similarities between the different types of instabilities for unbounded and bounded flows.

2.2 Unbounded flows

Interaction between the flow field and the surface can result in instabilities, based on flow and surface conditions. The surface which interacts with the flow field may be rigid or compliant. The growth of instability modes is likely the most important difference between the laminar-to-transient boundary-layer transition process over a rigid flat plate versus a compliant wall. Over a rigid flat surface transition is caused by Tollmien-Sclichting waves and is considered

an instability of the boundary-layer flow. Transition is modified by wall compliance. However, its basic behaviour remains the same. Instabilities of the compliant wall are considered wall-based and are characterized by the free wave modes which may become unstable at high flow speeds. They were termed flow induced surface instabilities by Carpenter and Garrad [CG85]. The two main types of flow induced surface instabilities are divergence (an essentially static instability) and traveling wave flutter. Hence, the instabilities in a system containing a compliant wall are flow based (Tollmien-Schlichting) or flow-induced surface instabilities. In addition, under certain circumstances, the Tollmien-Schlichting wave and traveling-wave flutter instabilities can interact to form a powerful new instability [CG85] [Wil86][CGW83] called a "transitional mode". The transitional mode has some of the characteristics of both of the flow based and flow-induced surface instabilities. The instabilities can also be classified according to whether they are convective or absolute. When the instabilities are convective [Gas62] [Gas65] (Tollmien-Sclichting waves and traveling wave flutter), the disturbance grows downstream from the source; the signal rises then decays. When instabilities are absolute (i.e. divergence and transitional mode), the signal rises but it does not decay. For this type of instability reducing the growth rate is not a viable strategy. The third kind of classification for the instabilities was termed by Benjamin [Ben60][Ben63] and Landahl [Lan62]. In this case, the instabilities were classified according to how they respond to irreversible energy transfer to and from the compliant wall. The instability can be characterized as Class A, B or C. Class A waves (i.e. Tollmien-Sclichting waves) are stabilized/destabilized by irreversible energy transfer to/from the compliant wall, whereas Class B waves (i.e. traveling wave flutter) responds in exactly the opposite way. Class C waves (i.e. divergence, transitional mode) are characterized by indifference to irreversible energy transfer. An important element of compliant wall research focuses on Class C waves, particularly on divergence. Class C waves represent a major concern, due to their instability characteristics.

Compliant surfaces have been shown to delay laminar-to-turbulent transition. Important the-

oretical studies on stability and transition in boundary-layers over flexible surfaces were performed by Benjamin [Ben60][Ben63], Landahl [Lan62], Landahl & Kaplan [LK65] and Gyorgyfalvy [Gyo67]. Their work were inspired by the so-called Gray's paradox [Gra36] (which concerns the unusually high swimming speed of dolphins), and the early experimental work of Kramer [Kra57] [Kra60a] [Kra60b] [Kra62], whose groundbreaking compliant coatings were modeled after the dolphin's epidermis. Kramer reported large drag reductions throughout his experiments and attributed these to transition-delaying properties of his compliant coating. However, as was pointed out later, he had an imperfect understanding of the structure and function of the dolphin's skin on which his compliant coating was based. Furthermore, the large drag reduction reported by Kramer was not verifiable and reproducible by other researchers. (In the 1980s it was demonstrated that the methods and facilities used by Kramer were unsuitable for studying transition-related phenomena.) Further experimental work has been carried out by Puryear [Pur62]. Major inconsistencies between the observation and the theories led to the abandonment of research until the 1980s, when new interest developed in the application of compliant walls for transition delay in water. Carpenter & Garrad [CG85] [CG86] demonstrated that in theory, substantial transition delays were possible using compliant walls. The laminar-to-turbulent transition over a flexible surface was investigated by Gaster [Gas87] and for turbulent boundary-layer interactions by Fischer *et al.* [FWBA75], Bushnell & Hefner [BH77], Dinkelacker [Din77] and Hansen & Hunston [HH74] [HH83]. Due to the large knowledge base on the transition-delaying process over flexible surfaces, the most recent research efforts offer optimizations of viscoelastic compliant walls. The limits of the transition-delaying capabilities of compliant walls made from viscoelastic materials for two compliant layers has been determined by Lucey, Carpenter & Dixon [CLD91][DLC94a][DLC94b]. Aspects of the boundary and end conditions for the two–fixed–ends configuration are shown by Lucey [Luc89]. Numerical simulations for the behaviour of a finite flexible plate in a uniform flow when continuous excitation is present was analyzed by Lucey [Luc98]. Although the present work concentrates on linear analysis, further insights about the more complex nonlinear behaviour

of flexible walls can be found in the article by Lucey *et.al.* [LCCY97].

2.3 Bounded or internal flows

2.3.1 FSI with commercial codes

The solution of fluid-structure interaction problems in the form of coupling of fluid and structural commercial programs are becoming very popular and viable through the accessibility of High Performance Computing. Typical examples are the coupling of CFX-5 and ANSYS [SP00] [PHS+00] for cardio-vascular applications or the coupling of in-house codes with 3D Spectrum [QTV98]. The coupling logic of the fluid and wall codes in this thesis is similar to the one used to couple CFX-5 and ANSYS. However, in the present analysis the two (self developed) codes are integrated and all calculations are performed internally.

While commercial codes are powerful, coupling them is often proved to be cumbersome. On the other hand self developed codes are more flexible and easy to tailor, but they may lack the generality and power of commercial CFD programs.

2.3.2 Benchmark geometries

To gain confidence in code predictions a validation exercise is required. The validation and case matrices are outlined in Section 1.3. The validation is based on unit and benchmarking cases. In the present analysis comparisons are made against lid-driven cavity flow and duct flow results for the fluid problems and against analytical solutions for the flexible plate cases. These unit testing cases provide simple and effective means to test the behaviour and prediction accuracy of computer codes.

Highly viscous lid-driven cavity flows for a triangular domain were studied experimentally by Moffatt [Mof64], demonstrating increasingly weaker connecting vortices. Numerical simula-

tions for square lid-driven cavity flows were performed by Rek & Škerget [RS94]. 2D and 3D lid-driven cavity flows were calculated by Teixeira [Tei97] for a wide range of Reynolds-numbers and compared with experimental results and against various commercial CFD code predictions. Here too, development of secondary corner vortices were demonstrated in both numerical and experimental studies.

Besides lid-driven cavity flows, duct flows are widely used for benchmarking purposes. Numerical simulations for simple Poiseuille type flows in a 2D duct, using ANSYS, are given in [dAl97] and in standard text books such as in [Bro67]. Backward facing steps are also very popular benchmark testing geometries to study flow characteristics. For example, numerical and experimental results were published by Gartiling [Gar90] using the finite element method. Freitas [Fre95] documented code comparison results from a number of commercial codes. Armaly *et.al.* [ADPS83] reported experimental and theoretical results for backward facing steps. Numerical simulations for both lid-driven cavity flows and backward facing steps are given by Bonfiglioli [Bon99].

Additional information on unit and benchmarking results and on basic test geometries can be found in standard textbooks, such as in [She90].

2.3.3 Flexible plates

The wall code, which models compliant plates in a 2D channel, is tested against analytical eigenmode solutions characterizing plate displacements. The eigenfunctions describing these modes are obtained from several sources.

A fundamental investigation of induced oscillations of a finite two-dimensional flexible plate considering unforced vibrations was presented by Walker *et.al.* [WZD97]. The induced oscillations of plates fixed at both ends were studied, documenting the corresponding eigenfunctions

and boundary conditions. Further work on the topic was done by Geveci [Gev99]. Transverse vibration of a rod was described by Nowacki [Now63], providing the eigenfunctions for cantilevered plates, clamped at one end and free at the other end.

Kornecki *et.al.* [KDO76] conducted theoretical and experimental investigations of flexible plates for both two–fixed–ends and cantilevered configurations. The flow was modeled as potential and incompressible and it was combined with a linear plate deformation model, producing divergence for the plate with two fixed trailing edges and flutter for the cantilevered configuration.

During divergence the aerodynamic forces and moments outgrow the restoring forces of the flexible plate. As the plate displacement increases the flow velocity also increases due to the reduced cross-sectional flow area. Consequently the aerodynamic forces increase too, leading to significant plate displacements. Divergence is considered a static instability, while flutter is regarded at as a dynamic instability. The latter is due to the coupling of the fluid dynamic forces and elastic inertial forces of the plate. The interaction between these forces can modify the vibration of the flexible plate. The resulting unsteady flow dynamics causes phase shifts between the plate displacement and the pressure as the plate extracts energy from the flow. Structural damping may influence this mechanism by slowing down the displacement growth or even reversing it. When the extracted energy from the flow to the structure is greater than the damping, the plate movement becomes divergent.

Yamaguchi *et.al.* [YYT00a] [YYT00b] proposed an analysis method for the prediction of fluttering of a flexible thin sheet or web swept by fluid flow. The flow moving in the x-direction was assumed two-dimensional and the plate oscillated with an infinitesimally small amplitude in the y-direction. The flow was approximated to be potential and the flow field is expressed in terms of distributed sheet vortices with an applied Kutta condition. These conditions do

not apply to fully developed flutter. Direct coupling of the fluid and wall configurations were used, instead of the superposition of characteristic modes and the time marching method employed by other researchers. The method was used in a small computer environment, just as in the present research. However, in the current analysis the Navier-Stokes equations are solved instead of the potential flow equations. It was also reported that at the high mass ratio region (mass ratio $\mu = \rho_{wall} h_{plate} / \rho_{fluid} L_{plate} > 0.7$) the oscillation modes and the frequencies were near to those of the second-order modes in vacuo, and the traveling waves were superposed on them. In the present analysis the initial wall displacement shape is also set to the second eigenmode.

Since the present research is concerned with the movement of the soft palate, a somewhat topical analysis is included in this review. A finite element model of the soft palate was developed by Berry *et.al.* [BMD99] using the commercial code, ABAQUS. The documented structural model outlined soft palate deformation time series while being in contact with the pharyngeal wall during snoring, providing another approach to examine the problem.

2.3.4 Upper airway studies

This thesis focuses on a particular application of fluid-structure interaction, namely on the dynamics of the upper airway. Details of the upper airway and its mechanics are complicated. Various aspects of its functions and dysfunctions have been studied through theoretical models and experiments and the results are published in open literature. In fact, flow stability in channels in connection with snoring and OSAH has become a problem of considerable interest during the past two decades. These medical conditions are modeled as flow over a flexible plate. In the upper airway two sections are affected by wall compliance. The first is the soft palate and the second is the pharyngeal region. The soft palate can be represented by a cantilevered elastic plate, while the lower section can be modeled by a flexible plate fixed at both ends. In this research linear analysis is adopted, since the interest lies in the initial stage of instability.

Through theoretical and experimental investigations Gavriely & Jensen [GJ93] proposed a lumped-parameter model that aimed at explaining the process of airway collapse. Their simple model used a plate-spring linear elasticity approach to model a movable wall in a channel section that connects to the airway opening through a narrowing resistance segment. The model predicted divergence and flutter for a wide range of conditions. The authors reported a snoring frequency between 30 and 100 Hz. The present analysis is performed at the upper (100 Hz) limit for reasons explained in Chapter 6.

The stability bounds and flutter were investigated theoretically and experimentally by Huang [Hua98] [Hua95] for flow over a cantilevered elastic plate in a two-dimensional channel (similar to the one employed here). Snoring (flutter) was modeled through linear analysis of the plate, where in vacuo modes, forming a series of orthogonal functions, were used to approximate the deformation of the fluid loaded plate. The incompressible inviscid flow was assumed uniform in the x-direction and satisfied the Kutta condition at the free trailing edge of the cantilevered plate. The numerical scheme employed a predictor-corrector method. In addition, Huang & Ffowcs Williams [HFW99] developed a mathematical model that coupled flow mechanics of the upper airway with the neurophysiological activity.

Theoretical studies of the fundamental characteristics of instability of plates in plain channel flow were performed by Guo & Paidoussis [GP00]. The unsteady inviscid potential-flow was assumed incompressible. The potential flow equations were solved using Fourier transform techniques. The plate deformation was accounted for through orthogonal beam functions. The plate was modeled using the Galerkin method and the study examined a wide range of boundary conditions for the leading and trailing edges of the plate in a two-dimensional channel flow. The mixed fluid-plate interaction boundary condition lead to a singular integral equation. The loading and trailing edge conditions resulted in divergence and flutter. In their work the au-

thors used similar geometries to the one studied here.

Theoretical and experimental studies of the soft palate aeroelastic instability were conducted by Aurégan & Depollier [AD95], using incompressible inviscid unsteady potential flow and orthogonal in vacuo beam functions for plate deformation. The length of the vibrating plate was found to be the same order as the transverse vibration wavelength. Only the first two modes of vibration of the soft palate were taken into account. Both theory and experiments showed that the plate lost stability in the form of flutter. They recommended a non-linear model to explain the complete motion of the soft palate, and the large abrupt changes in the pressure, resulting from the closing of the pharyngeal channel which lead to sound production.

Luo & Pedley [LP00] performed numerical investigations to study the flow limitation phenomenon in a 2D collapsible channel, where a wall segment on one side of the channel was made compliant. During flow limitation the upstream transmural pressure is held constant. The analysis was performed under steady and unsteady flow conditions using various boundary conditions. Throughout the calculations a Poiseuille type inlet or outlet velocity profile was assumed, depending on the boundary conditions. Previous work ([LP98], [LP96], [LP95]) used steady flow conditions, modeling the pressure limitation phenomenon, where the downstream transmural pressure is held constant. A second-order predictor-corrector implicit difference scheme was used with variable time discretization controlled by error tolerance to solve the time dependent coupled problem. The finite element method for the fluid flow, employed quadratic velocity and linear pressure nodes. The non-linear equations for the elastic wall were solved using the Newton-Raphson method. Although the geometries are somewhat different, there are several similarities between the works of Luo & Pedley and the research reported here. For example, in both cases the Navier-Stokes equations were solved using the FEM method, with triangular elements, quadratic velocity nodes and linear pressure nodes. Inlet velocity and a zero outlet pressure were given as boundary conditions. However, the present model uses

an explicit FEM formulation for the fluid, a cantilevered plate in the middle of the channel, uniform inlet velocity and two to four times fewer elements during domain discretization.

Besides the rectangular duct geometry a demonstration run is performed using axially varying channel diameters, corresponding to the vocal tract between the oral inlet and the vocal cords. These geometries were derived from the work of Dedouch *et.al.*[DHV+99].

Without further details, significant contribution to the behaviour and stability modeling of collapsible tubes under steady and unsteady conditions were given by Bertram [Ber82] [Ber86], Bertram & Pedley [BP82] and Bertram & Raymond [BR90]. These papers will be relevant in future studies, where a compliant pharyngeal region is modeled further to the cantilevered soft palate.

2.4 Motivation for the research

As outlined in Chapter 1, the motivation for the current research is to develop a stable computational method suitable to study coupled fluid-structure interaction scenarios, with complex internal flow geometries and to use the resulting fully coupled code to gain insight into upper airway instability mechanisms. Although many aspects of the fluid behaviour for internal flows and elastic plate deformation/dynamics were studied before and reported in open literature, it was found that the geometries and configurations proposed for this thesis were not yet explored to this complexity. In particular, the approach and the numerical scheme used here is similar to the one employed by Luo & Pedley ([LP98], [LP96], [LP95]), with the important difference that their compliant wall section was fixed at both ends. The current model of a split-inlet 2D channel with a flexible cantilevered section resembles more the geometries used by Huang [Hua98] [Hua95] and Guo & Paidoussis [GP00]. While these authors used potential-flow models and the plate movement was restricted, the present research will examine the fluid-structure

interaction in a two-dimensional channel in a viscous flow field, using incompressible fluid at low Reynolds-numbers, where the flexible plate deformation is not prescribed and feedback mechanisms between the flexible plate and the fluid are permitted. Only the second eigenmode of the initial plate displacement is considered. The pressure variation induced by the motion acts on both sides of the flexible plate – even when one inlet is blocked – with zero pressure difference at the free end of the cantilevered flexible plate, thereby automatically satisfying the Kutta condition.

The Navier-Stokes equations are solved using an explicit FEM formulation. Displacements of the flexible plate are calculated using an implicit finite difference method. These self developed codes, in conjunction with fast mesh generators for various geometries, are fully coupled to solve the fluid-structure interaction problems at hand.

This study adds to our knowledge base with respect to the numerical method developed and regarding the instability mechanisms for complex upper airway geometries, namely flutter and divergence.

Chapter 3

Theory and computational models

This chapter outlines the theory used to calculate the fluid velocity and pressure distribution, plate displacement and the flow field for a coupled model.

The fluid section starts with the general Navier–Stokes equations, then the initial and boundary conditions are introduced. The Finite Element Method is explained in detail. The key equations used to obtain the weak and the block forms of the Navier–Stokes equations are shown. The elemental and global constructions with programing logics for code development, mesh generation, solution techniques and other relevant information for the FEM method are given in Appendix E. The flexible plate section introduces the basic equations of motion and the initial and boundary conditions for various plate end conditions.

Finally, the last section outlines the method used to couple the fluid and wall codes.

3.1 The Navier–Stokes equations

By looking at sub-components of the Navier-Stokes equations, the Poisson equation and the steady and unsteady Stokes equations can be written. The Poisson equation, which includes the diffusion term, is written in the form of

$$-\nabla^2 \mathbf{u} = f \text{ in } \Omega \tag{3.1}$$

$$\mathbf{u} = 0 \text{ on } \partial\Omega$$

where f is the forcing term, and assuming "no slip" ($u = 0$) at the boundaries.

When the flow in the domain is considered highly viscous, the motion of this type of fluid can be described using the Stokes equations. Adding the pressure force to Equation 3.1 and writing it with the continuity equation, the Stokes equations can be given as

$$-\nabla^2\mathbf{u} + \nabla\mathbf{p} = f \text{ in } \Omega \tag{3.2}$$

$$\nabla\mathbf{u} = 0 \text{ in } \Omega \tag{3.3}$$

$$\mathbf{u} = 0 \text{ on } \partial\Omega$$

here the pressure force includes the dynamic viscosity (i.e., $\mathbf{p} = \frac{p}{\mu}$).

The unsteady Stokes problem includes a time dependent term ($\partial\mathbf{u}/\partial t$)

$$\frac{\partial\mathbf{u}}{\partial t} = \nabla^2\mathbf{u} - \nabla\mathbf{p} + f \text{ in } \Omega \tag{3.4}$$

$$\nabla\mathbf{u} = 0 \text{ in } \Omega$$

$$\mathbf{u} = 0 \text{ on } \partial\Omega$$

The motion of the fluid can be described using the Navier–Stokes equations. The convection term ($\mathbf{u} \cdot \nabla\mathbf{u}$) is added to Equation 3.4, which is non-linear (from $u \cdot u$), non-symmetric, and anisotropic. Modeling the flow as incompressible, the complete mathematical statement of these equations in 2D (if the heat transfer is neglected), is to solve three equations. The first two equations below account for the conservation of momentum and the third for the steady state conservation of mass at every point of the two dimensional flow field.

$$\frac{\partial u_1}{\partial t} + \left(u_1\frac{\partial u_1}{\partial x} + u_2\frac{\partial u_1}{\partial y}\right) = -\frac{\partial p}{\partial x} + \frac{1}{Re}\left(\frac{\partial^2 u_1}{\partial x^2} + \frac{\partial^2 u_1}{\partial y^2}\right) + f \tag{3.5}$$

$$\frac{\partial u_2}{\partial t} + \left(u_1\frac{\partial u_2}{\partial x} + u_2\frac{\partial u_2}{\partial y}\right) = -\frac{\partial p}{\partial y} + \frac{1}{Re}\left(\frac{\partial^2 u_2}{\partial x^2} + \frac{\partial^2 u_2}{\partial y^2}\right) + f \tag{3.6}$$

$$\frac{\partial u_1}{\partial x} + \frac{\partial u_2}{\partial y} = 0 \tag{3.7}$$

The Navier–Stokes equations given in Equations 3.5, 3.6 and 3.7 for unsteady non-creeping flow in their dimensionless form can be re-written as

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = \frac{1}{Re} \nabla^2 \mathbf{u} - \nabla \mathbf{p} + f \text{ in } \Omega \tag{3.8}$$

$$\nabla \cdot \mathbf{u} = 0 \text{ in } \Omega \tag{3.9}$$

for three unknowns (u_1, u_2, p), subject to the following boundary conditions:

At the solid wall (at $\partial\Omega$)

(i)	No slip	$u_1 = 0$
(ii)	Impermeability	$u_2 = 0$

Infinitely far from the solid wall in both x and y directions

(iii)	Uniform flow	$u_1 = U_\infty$
(iv)	Uniform flow	$u_2 = 0$

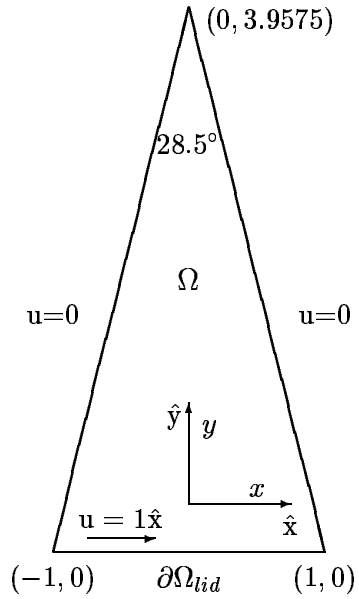
It should be also noted that the pressure in this formulation is only determined up to a constant $(u, p + C)$. To obtain a unique pressure solution, an additional condition should be included, such as

$$\int_\Omega p \, dx = 0 \tag{3.10}$$

The flow considered has the following properties [CSvS86]:

1. The medium is incompressible
2. The medium is Newtonian
3. The medium properties are temperature independent and uniform

Figure 3.1: Physical Triangular Lid–Driven Cavity Domain



4. The flow is non-turbulent

The above equations are used to calculate velocity and pressure conditions in various two-dimensional domains. The code developed for this study does not account for gravity or heat addition. The Navier–Stokes equations are formulated as dimensionless equations, where the fluid density and viscosity are accounted for through a suitable Reynolds-number. In the building block approach to computer code validation (see Appendix C) two main domain configurations are used: lid–driven cavity flows and duct flows. Examples of these unit and benchmarking domains (Ω) are shown in Figures 3.1 and 3.2 for lid–driven cavity flows (triangular and square domains), and in Figure 3.3 for generic 2D duct flows.

Figure 3.2: Physical Rectangular Lid–Driven Cavity Domain

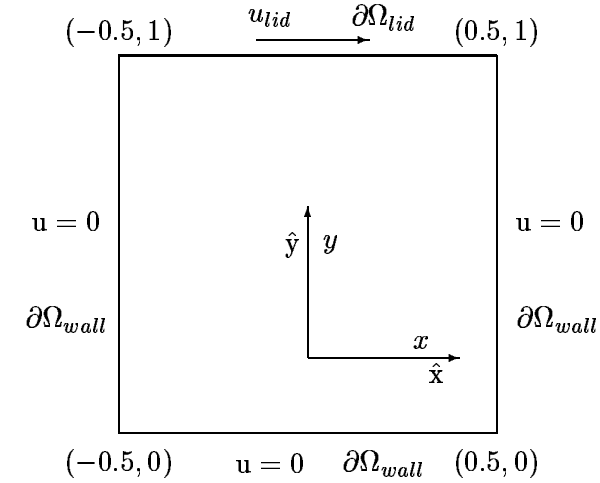
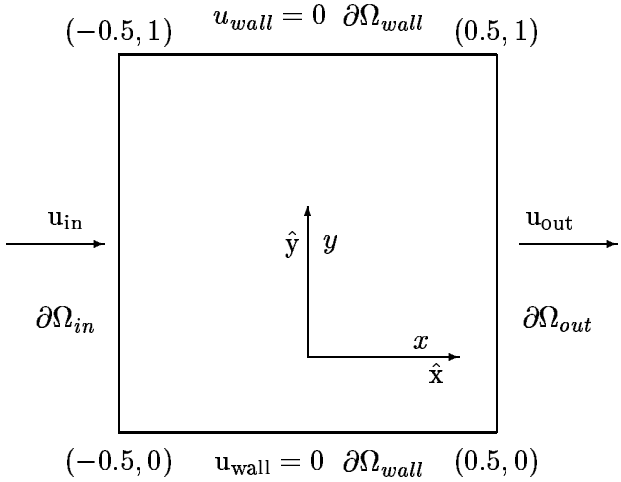


Figure 3.3: Physical 2D Duct Domain



3.2 Initial and boundary conditions for internal flows

3.2.1 For lid–driven cavity flows

Based on the no-slip boundary condition, velocities at the non moving wall boundary nodes are zero. Along the sliding lid boundary nodes the velocity boundary conditions are given parallel to the lid and zero normal to the lid, with boundary conditions

$$\begin{aligned} \mathbf{u} &= \mathbf{C}\hat{\mathbf{x}} \quad \text{on } \partial\Omega_{\text{lid}} \\ \mathbf{u} &= 0 \quad \text{on } \partial\Omega \setminus \partial\Omega_{\text{lid}} \quad \text{(everywhere but lid)} \end{aligned}$$

and initial conditions

$$\mathbf{u}(\mathbf{x}, t = 0) = 0$$

where Ω represents the domains shown in Figures 3.1 and 3.2. ($\mathbf{C}$ is a scaling constant.)

For triangular and rectangular lid–driven cavity cases, the velocity, length, time and pressure are scaled by U_{lid} (the dimensional lid velocity), ℓ (half the dimensional lid extent), ℓ/U_{lid}, and ρU_{lid}^2 respectively. The Reynolds number is $Re = \rho U_{\text{lid}} \ell/\mu$, where ρ and μ are density and dynamic viscosity, respectively.

3.2.2 For 2D duct flows with steady inlet flow conditions

For the 2D duct flow cases Dirichlet – or essential – boundary conditions are used, where the dependent variables u and v are specified along the boundaries.

Velocities at the wall boundary nodes are zero due to the no-slip boundary condition. These are the top and bottom sides of the rectangular "duct" domain. The no-slip condition also applies to split domains, where the wall velocities at the wall splitting the domain are zero.

The inlet velocity profile can be assumed uniform or non uniform. When a simpler uniform velocity profile is used, the flow velocity values parallel to the flow direction are set to a constant

value for each node at the inlet boundary. The normal flow component (relative to the flow direction) is set to zero. Here this inlet condition is used, based on the assumption that during breathing the upper airway geometries would not allow for the development of a parabolic velocity profile.

For fully developed flows, the velocity profile is parabolic. An example for this is the Poiseuille flow or Hagen–Poiseuille flow. At the inlet the velocity profile is parabolic and the inlet boundary conditions can be defined as

$$\begin{aligned} u_{1in} &= U_{x,max}\left(1-\left(\tfrac{y}{Y}\right)^2\right) && \text{on } \partial\Omega_{\text{in}} \\ u_{2in} &= 0 && \text{on } \partial\Omega_{\text{in}} \\ \mathbf{u} &= 0 && \text{on } \partial\Omega \setminus \partial\Omega_{\text{in}} \quad \text{(everywhere but inlet)} \end{aligned} \tag{3.11}$$

where y is the distance from the center of the duct and Y is the half height of the duct. (In 3D this would represent the distance from the center of the pipe and the radius of the pipe.) u_1 is the velocity component in the x direction. The parabolic profile can be scaled by the $U_{x,max}$ multiplier, representing the maximum velocity in the x direction.

The initial condition is defined as zero velocity at $t = 0$

$$\mathbf{u}(\mathbf{x}, t = 0) = 0 \tag{3.12}$$

For 2D duct flow cases, the velocity, length, time and pressure are scaled by U_{in} (the peak dimensional inlet velocity), $2D$ (twice the dimensional duct height), $2D/U_{\text{in}}$, and ρU_{in}^2 respectively. The Reynolds-number is $Re = \rho U_{\text{in}} 2D/\mu$, where ρ and μ are density and dynamic viscosity, respectively.

The BC's for 2D duct flows with unsteady inlet flow conditions are given in Appendix D.

It should be noted that the use of a steady inlet flow condition is a simplification of the problem. In reality the flow is pressure driven. For a given constant pressure drop between the

inlet and the outlet of the duct, the inlet velocities may change, based on the flow resistance in the channel. This issue will be further discussed in Section 6.3.

3.2.3 Outlet conditions for 2D duct flows

The outlet velocity in the y direction (u_2) is assumed to be zero. That is, at the outlet, only the x velocity components exist.

$$u_{2out} = 0 \quad \text{on} \quad \partial\Omega_{\text{out}} \tag{3.13}$$

3.3 Finite Element Method for incompressible flows

The finite element method (FEM) is one of the most commonly used method for solving partial differential equations (PDEs). Discretizations by FEMs (or Finite Difference Methods (FDMs)) transform the Partial Differential Equations (PDEs) into a system of linear algebraic equations.

It makes use of the computer and is very general in the sense that it can be applied to:

- Steady-state or unsteady conditions;
- Linear or nonlinear problems;
- Geometries of arbitrary space and dimension.

The finite element method (FEM) solves variational forms of a differential equation in a piecewise manner over each element in the domain. The order of each element is defined by the degree of polynomial, used by that element to approximate the primary variables resulting in some equation. These element equations may then be assembled together so that the global stiffness and mass matrices and the forcing vector are obtained. Boundary conditions are then imposed on the matrix and a solution for the primitives may be obtained.

The major steps in the finite element analysis of a typical problem are:

1. Discretization of the domain into a set of finite elements (mesh generation)
2. Weak formulation of the differential equation to be analyzed
3. Development of the finite element model of the problem using its weak form
4. Assembly of finite elements to obtain the global system of algebraic equations
5. Imposition of boundary conditions
6. Solution of equations
7. Post-processing of solution and quantities of interest.

A basic methodology of discretization, weak formulation and the development of the finite element model of the present problem are outlined here. Practical aspects of the FE method used, including the block formation process for elemental and global matrix constructions, mesh generation, solution methods and related issues are further described in Appendix E.

In addition, detailed mathematical description of the method can be found in many text books, for example in [RG94], [Bat96] and [Red93].

3.3.1 Discretization of the domain

The domain Ω is partitioned into a finite number (K) of subdomains $\Omega_1, \Omega_2, \ldots, \Omega_K$, called *finite elements*. These elements are non-overlapping and two-dimensional in this case, where the boundary Γ of Ω is made up of straight segments, covering the entire domain with polygonal elements. Every side of the element is either part of the boundary Γ or a side of another element.

Certain points are identified in the subdomain, called *nodes* or *nodal points*. Nodes are allocated at least at the vertices of elements (linear elements), but in order to improve the approximation, further nodes may be introduced, for example at the mid-points of the sides of elements

(quadratic elements). These nodes, numbered $1, 2, \ldots, G$, have position vectors $x_1, x_2, \ldots, x_G$. The set of elements and nodes that make up the domain Ω is called a *finite element mesh.*

Taylor–Hood elements

Historically one of the most important group of elements for approximating the Navier-Stokes equations is the so-called "Taylor-Hood" group of elements and they remain widely used. One of the first successful finite element approximation was calculated by Hood and Taylor [TH73] using continuous, piecewise quadratic velocities and continuous, piecewise linear pressures on a grid of triangular elements. Currently the name "Taylor-Hood" element is used to describe any finite element approximation using the same grid for pressure and velocity, which provide continuous approximations for velocity and pressure based on Lagrange type interpolation, where the order of approximation is one less for pressure than for the components of velocity. In practice, the only elements that are widely used are based on quadratic velocities and linear pressures or tetrahedra and triquadratic velocities and trilinear pressures on hexahedra and it is the latter that are most often used for three-dimensional flow. In the present work the FEM formulation of the Stokes and Navier-Stokes equations for two-dimensional channel flow are solved by Taylor-Hood ($\mathbb{P}_2 - \mathbb{P}_1$) formulation. These elements are characterized by the fact that the pressure is continuous in the region of Ω. The velocity is quadratic, the pressure is linear. Due to the choice of the nodal points, both velocity and pressure are continuous over the elements boundaries. For example, for the pressure, the linear approximation as well as the use of the vertices fix the pressure along the boundary, once the nodal point values are given. Since for adjacent elements, the joint side has the same nodal points and the same linear approximation, the pressure is continuous on this boundary in both elements, and therefore it is continuous in Ω.

3.3.2 Basis functions

In the next step the basis (or shape) functions (N_i) are introduced. These functions are bounded and continuous. There is one function for each node (G number of nodes) and each function N_i is non-zero only on those elements that are connected to node i. N_i is equal to 1 at node i and equal to zero at the other nodes. $N_i^{(k)}$ is called a local basis (or shape) function. The basis (or shape) functions are set up as piecewise polynomials and they are non-zero only in a "small" region. A typical basis N_i (called a global basis function) is built up by patching together the local basis functions $N_i^{(k)}$ associated with node i. The function v can be approximated with the function v_h using these basis functions:

$$v_h(x) = \sum_{i=1}^{G} b_i N_i(x) \tag{3.14}$$

as a consequence of

$$N_i(x_j) \begin{cases} 1 \text{ if } i = j \\ 0 \text{ otherwise} \end{cases} \tag{3.15}$$

the coefficient b_j below is simply the value of v_h at node j

$$v_h(x_j) = \sum_{i=1}^{G} b_i N_i(x_j) = b_j \tag{3.16}$$

By patching together the functions, it can be seen that each basis function $N_i(x)$ is a piecewise linear "hat" function made up of the local basis functions associated with the node i. Hence every function v_h is a piecewise linear function in which b_i is the value of v_h at node i.

3.3.3 The Galerkin method

The basic idea behind the Galerkin method is an extremely simple one [Red98]. Consider the variational boundary value problem of finding $u \in V$ that satisfies

$$a(u, v) = \langle l, v \rangle \qquad \text{for all } v \in V \tag{3.17}$$

where V is a subspace of a Hilbert space H. It is possible to pick a few linearly independent functions $\Phi_1, \Phi_2, \ldots, \Phi_N$ in V and define the space V_h to be the finite-dimensional subspace of V spanned by the functions Φ_i. That is

$$V^h \subset V, \qquad \text{span}\{\Phi_i\}_{i=1}^N = V^h \tag{3.18}$$

The index h is a parameter that lies between 0 and 1, and whose magnitude gives some indication of how close V^h is to V; h is related to the dimension of V^h, and as the number N of basis functions chosen gets larger, h gets smaller (for example, one can set $h = 1/N$). In the limit, as $N \to \infty$, $h \to 0$ and $\{\Phi_i\}$ is chosen is such a way that V^h will approach V.

Having defined V^h, problem 3.17 is now posed in V^h instead of in V. That is, a function $u_h \in V^h$ is sought that satisfies

$$a(u_h, v_h) = \langle l, v_h \rangle \qquad \text{for all } v_h \in V^h \tag{3.19}$$

This is the essence of the Galerkin method. In order to solve for u_h, it can be noted that both u_h and v_h must be linear combinations of the basis functions of V^h, so that

$$u_h = \sum_{i=1}^{N} c_i \Phi_i \qquad \text{and} \qquad v_h = \sum_{j=1}^{N} d_j \Phi_j \tag{3.20}$$

Since v_h is arbitrary, so are the coefficients d_j. Substitution of Equation 3.20 in Equation 3.19 and use of the fact that a is bilinear and l is linear, lead to the equation

$$\sum_{i=1}^{N} \sum_{j=1}^{N} a(\Phi_i, \Phi_j) c_i d_j = \sum_{j=1}^{N} (l, \Phi_j) d_j \tag{3.21}$$

or more concisely

$$\sum_{j=1}^{N} d_j \left(\sum_{i=1}^{N} K_{ij} c_i - F_j \right) = 0 \tag{3.22}$$

in which

$$K_{ij} = a(\Phi_i, \Phi_j) \qquad \text{and} \qquad F_j = \langle l, \Phi_j \rangle \tag{3.23}$$

Note that K_{ij} and F_j can be evaluated in practice since the Φ_i are known functions and the forms of a and l are also known. Since the coefficients d_j are arbitrary, it follows that Equation 3.22 only holds if the term in brackets is zero. The problem is thus reduced to one of solving the set of *simultaneous linear equations*

$$\sum_{i=1}^{N} K_{ij}\ c_i = F_j \qquad\qquad j = 1, \ldots, N \tag{3.24}$$

or more compactly

$$\mathbf{K}^t\mathbf{c} = \mathbf{F} \tag{3.25}$$

in which $\mathbf{K}$ and $\mathbf{F}$ are, respectively, the *stiffness matrix* and *load* or *forcing vector* with entries K_{ij} and F_j. Once these equations are solved, the approximate solution u_h can be found from the first of equations 3.20.

3.3.4 Weak form of the Navier–Stokes equations

The strong formulation requires evaluation of the integral which includes the highest order of derivative term in the differential equation (see Equations 3.5, 3.6 and 3.7). The integral must have non-zero finite values to yield a meaningful approximate solution to the differential equation. This means that a trial function should be differentiable twice and its second derivative should not vanish. So as to reduce the requirements for a trial function in terms of order of differentiability, integration by parts is applied to the strong form formulation. The trial function needs the first order differentiation instead of the second order differentiation. As a result, the requirement for the trial function is reduced. This formulation is called the *weak formulation.*

The Navier-Stokes equations and the continuity equation must be solved simultaneously, because together they describe the flow. Unknowns in these equations are the velocity components and the pressure. (The work in this thesis is limited to incompressible flow.)

Starting from the Navier-Stokes equations and the continuity equation, and applying the Galerkin method, basis functions for velocities and pressure are used as test functions. This leads to a system of equations in matrix form with velocity values and pressure values at the nodal points as unknowns. Velocity and pressure couple simultaneously.

Note that the solution of the pressure from the Navier-Stokes equations is determined up to an additive constant, when nowhere normal stresses have been prescribed on the boundary. One way to fix this constant is by satisfying Equation 3.10. Of course this condition can be discretized, but in numerical computation for the sake of simplicity one usually prescribes the pressure at one nodal point.

The Navier–Stokes equations, expressed solely in terms of the velocity components and pressure can be summarized as

$$\frac{\partial u_i}{\partial t} + u_j \frac{\partial u_i}{\partial x_j} - \frac{\partial}{\partial x_j}\left[-p + \frac{1}{Re}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)\right] + f = 0 \tag{3.26}$$

$$\frac{\partial u_i}{\partial x_i} = 0 \text{ in } \Omega \tag{3.27}$$

The finite element model is a natural formulation in which the weak forms of the above equations are used to construct the finite element model. The resulting finite element model is termed the *velocity–pressure model* or *mixed model*. The expression "mixed" is used because velocity variables are mixed with the force-like variable, pressure, and both types of variables are retained in a single formulation.

The starting point for the development of the finite element models of Equations 3.26 and 3.27 is their weak forms. After adding the weight functions (w and q) – which will be equated, in the Rayleigh-Ritz-Galerkin finite element models, to the interpolation functions used for $(p, \mathbf{u})$, respectively – the weak form or the equations are obtained by integration-by-parts. This equally

distributes integration between the dependent variables and the weight functions. However, in any problem, such trading of differentiability is subjected to the restriction that the resulting boundary expressions are physically meaningful. Otherwise, the secondary variables of the formulation may not be the quantities the physical problem admits as the boundary conditions.

After integrating by parts, the weak form of the Navier–Stokes equations are written as

$$0 = \int_{\Omega^e} \left[\left(w_i \frac{\partial u_i}{\partial t} + w_i u_j \frac{\partial u_i}{\partial x_j} \right) + \frac{\partial w_i}{\partial x_j} \left(-p + \frac{1}{Re} \left\{ \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right\} \right) - w_i f_i \right] d\mathbf{x} \quad (3.28)$$

$$0 = \int_{\Omega^e} q \left(\frac{\partial u_i}{\partial x_i} \right) d\mathbf{x} \quad (3.29)$$

3.3.5 Block form of the Navier–Stokes equations

The weak form of the Navier–Stokes equations is given in Section 3.3.4. However, the weak formulation is only half of the Galerkin method. The other half is the introduction of the appropriate trial solution. This subsection outlines the key equations used to build the Navier–Stokes equations in block form. After the discretization of the unsteady Stokes equation, the convection term is defined. Combining these equations results in the formulation used to model velocity and pressure conditions in a two-dimensional domain. Detailed description of the method is given in Appendix E.

The aim is to find trial functions as approximate solutions for velocity and pressure for every time step $t \in]0, T[$ such that

$$u_h(x,t) \in x_h \subset x \quad (3.30)$$

$$P_h(x,t) \in M_h \subset M \quad (3.31)$$

Using the Galerkin formulation with Taylor–Hood ($\mathbb{P}_2$ — $\mathbb{P}_1$) element bases

$$u_{ih}(x,t) = \sum_{n=1}^{N^V} (u_i)_n(t) \Phi_n(x) \quad (3.32)$$

$$P_h(t) = \sum_{m=1}^{N^P} P_m(t)\Psi_m(x) \tag{3.33}$$

where N^V and N^P are the number of velocity and pressure nodes.

Unsteady Stokes equation

Starting with the FEM discretization of the unsteady Stokes equation

$$\rho\, w\, \frac{\partial u_h}{\partial t} = \frac{\partial w}{\partial x}\, P_h - \mu \left(\frac{\partial w}{\partial x} \cdot \frac{\partial u_h}{\partial x} \right) + wf \qquad \text{for all } w \in X_h \tag{3.34}$$

$$\frac{\partial u_h}{\partial x}\, q = 0 \qquad \text{for all } q \in M_h \tag{3.35}$$

The semi-discrete algebraic equation is

$$Mu_t = D^T P - Au + Mf \tag{3.36}$$

$$Du = 0 \tag{3.37}$$

with $u(t=0) = u_0$, where the mass matrix M is an $N \times N$ matrix

$$M_{ij} = \int_\Omega \Phi_i \Phi_j dx \qquad (i,j) \in (0, \ldots, K-1)^2 \tag{3.38}$$

where K is the number of elements in the domain.

The stiffness matrix A is also an $N \times N$ matrix

$$A_{ij} = \int_\Omega \frac{d\Phi_i}{dx} \frac{d\Phi_j}{dx} dx \tag{3.39}$$

For the temporal discretization the Euler Forward method is used

$$M \frac{u^{n+1} - u^n}{\Delta t} = D^T P^{n+1} - Au^{n+1} + Mf^{n+1} \tag{3.40}$$

$$Du^{n+1} = 0 \tag{3.41}$$

the forcing term is given, and known all the time. It is a term independent from velocity and pressure. For the cases examined in the present work this forcing term is neglected.

Convection term of the Navier–Stokes equations

For the forcing term neglected, ($f = 0$) the convection term can be discretized as

$$M\frac{du}{dt} = Cu \tag{3.42}$$

where M is the mass matrix (see Equation 3.38) and C is the convection matrix

$$C = -U\int_{\Omega}\Phi_i\frac{d\Phi_j}{dx}dx \qquad (i,j) \in (0,\ldots,K-1)^2 \tag{3.43}$$

using the lumped matrix for explicit schemes

$$\frac{du}{dt} = \frac{1}{h}Cu \tag{3.44}$$

with the initial condition $u(t=0) = u_0$.

Block form of the Navier–Stokes equations

The weak form of the Navier–Stokes equations are given as

$$\rho\ w\ \frac{\partial u_h}{\partial t} + \rho\ w\ u_h\ \frac{\partial u_h}{\partial x} = \frac{\partial w}{\partial x}\ P_h - \mu\left(\frac{\partial w}{\partial x}\cdot\frac{\partial u_h}{\partial x}\right) + wf \qquad \text{for all } w \in x_h \tag{3.45}$$

$$\frac{\partial u_h}{\partial x}\ q = 0 \qquad \text{for all } q \in M_h \tag{3.46}$$

The block form of the Navier–Stokes equations is

$$M\frac{du}{dt} + C(u)u = D^T P - Au + Mf \tag{3.47}$$

$$D\cdot u = 0 \tag{3.48}$$

The convection matrix $C(u)$

$$C(u) \equiv \left(w, \sum_{\beta=1}^{2}(u_\beta)\frac{\partial w}{\partial x_\beta}\right) \tag{3.49}$$

after inserting the bases

$$C(u) \equiv \left(\Phi_i, \sum_{\beta=1}^{2}\sum_{n=1}^{N^V}(u_\beta)_n\Phi_n(x)\sum_{j=1}^{N^V}w_j\frac{\partial\Phi_j}{\partial x_\beta}\right) \tag{3.50}$$

after rearranging the matrix

$$\sum_{j=1}^{N^V}\left(\int_\Omega \Phi_i(x)\sum_{\beta=1}^{2}\sum_{n=1}^{N^V}(u_\beta)_n\Phi_n(x)\frac{\partial\Phi_j}{\partial x_\beta}dx\right)w_j \tag{3.51}$$

from which the convection matrix $(N^V \times N^V)$is

$$C_{ij}=\sum_{\beta=1}^{2}\sum_{n=1}^{N^V}(u_\beta)_n\int_\Omega \Phi_i(x)\Phi_n(x)\frac{\partial\Phi_j}{\partial x_\beta}dx \tag{3.52}$$

The absolute stability and the Adams–Bashforth 3 numerical scheme used for the discretization of the convection term is outlined in Section 3.3.6.

Based on (a) the block form derivation of the Stokes equations, (b) the time discretization using the conditionally absolute stable Euler Forward scheme and (c) the AB3 based convection term, the final block form of the Navier–Stokes equations can be given as

$$\begin{bmatrix} \frac{1}{Re}A+\frac{1}{\Delta t}M & 0 & -D_1^T \\ 0 & \frac{1}{Re}A+\frac{1}{\Delta t}M & -D_2^T \\ -D_1 & -D_2 & 0 \end{bmatrix}\begin{Bmatrix} u_1^{n+1} \\ u_2^{n+1} \\ p^{n+1} \end{Bmatrix} = \frac{1}{\Delta t}\begin{bmatrix} M & 0 & 0 \\ 0 & M & 0 \\ 0 & 0 & 0 \end{bmatrix}\begin{Bmatrix} u_1^n \\ u_2^n \\ 0 \end{Bmatrix} \tag{3.53}$$
$$-\frac{1}{12}(23C^n-16C^{n-1}+5C^{n-2})$$

where the convection term components, $C^q(q=n,\ n-1,\ n-2)$, are obtained from direct-stiffness summation of

$$\sum_{\delta=1}^{6}\left(\hat{C}^k_{\alpha\delta}\right)^q u^q_{g^k_\delta} \tag{3.54}$$

over k and α. The time step (Δt) used in the calculations is set only by the convection term.

3.3.6 Stability implementation for the convection term

In the Navier–Stokes equations the convection term is non-linear. To achieve absolute stability, explicit or implicit numerical techniques can be chosen. Absolute stability is defined as a property by which the numerical solution of a physically stable problem would be bounded, irrespective of the time step. The advantages and disadvantages of explicit schemes are given

below. The advantages and disadvantages of an implicit scheme will be given in Section 3.4.

The explicit approach is relatively simple to set up and program. Therefore, it was chosen for the present FEM discretization scheme. The disadvantage is that, for a given Δx, the time step (Δt) must be less than some limit imposed by stability constraints. In some cases the time step must be very small to maintain stability. This can result in long computer running times to make calculations over a given interval of t.

The Adams-Bashforth (AB) family belongs to the explicit schemes. The stability boundary for the first three AB methods is shown in Figure 3.4. (Euler Forward is AB1.) The curves plot the real and imaginary portions of $\lambda\Delta t$ (where λ is an eigenvalue) for the ODE $\dot{y} = \lambda y$ for complex λ, and each method is stable only when $\lambda\Delta t$ lies on (the stability limit) or within the closed stability boundary. They are obtained by seeking the solution of the $y_n = y_0\xi^n$ and setting $\xi = e^{i\theta}$, which has $|\xi| = 1$. For example, applied to explicit Euler, which can be derived as $y_{n+1} = (1 + \lambda\Delta t)y_n = \xi y_n = e^{i\theta}y_n$ or $\lambda\Delta t = e^{i\theta} - 1 = (cos\ \theta - 1) + i\ sin\ \theta$, which, as θ ranges from 0 to 2π, describes the curve labeled Euler Forward. For other stability curves see [GS00]. (Note that for Euler Forward the circle ranges between $-2 < x < 0$ and $-1 < y < 1$. For AB2 and AB3 the x values are $-1 < x < 0$ and $-0.5 < x < 0$, respectively.)

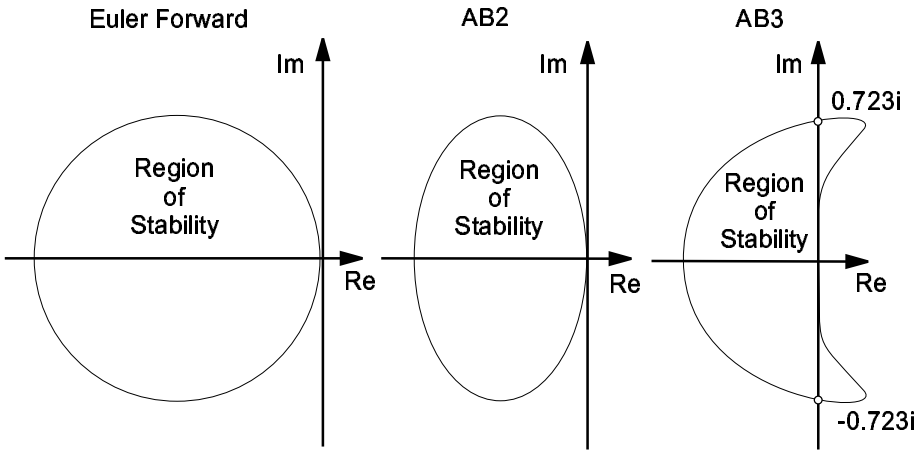
The AB3 scheme is conditionally stable. For stability the time step restriction (Δt) is scaled to get into the positive region (CFL condition)

$$\Delta t\frac{u}{h} \leq \Delta t_{cr}\frac{u}{h} = 0.723 \tag{3.55}$$

$$h \to 0 \qquad \Delta t_{cr} \to 0 \tag{3.56}$$

There is a linear relationship between the element size (h) and the critical time step (Δt_{cr}). For example, if the element size decreases by a factor of 2 the critical time step also decreases by a factor of 2.

Figure 3.4: Adams-Bashforth family for stability [Par99]



Adams-Bashforth 3 method [GS00] is implemented as follows:

$$\frac{u^{n+1}-u^n}{\Delta t} = \frac{1}{h}\left(\frac{23}{12}Cu^n - \frac{16}{12}Cu^{n-1} + \frac{5}{12}Cu^{n-2}\right) \tag{3.57}$$

the stability criteria can be expressed as

$$\Delta t \leq \Delta t_{cr} = \min\left(\frac{h'}{u'}\right) \cdot 0.723 \cdot \mathrm{SF} \tag{3.58}$$

where SF is a safety factor (e.g., SF=0.9) and

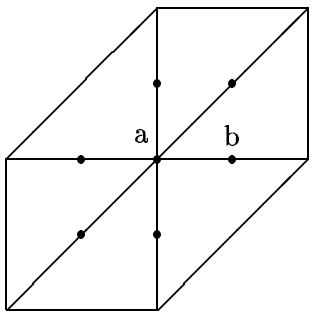
$$h'_b = |X_b - X_a| = \sqrt{(x_b - x_a)^2 + (y_b - y_a)^2} \tag{3.59}$$

$$u'_b = \frac{|U_b \cdot (X_b - X_a)|}{|X_b - X_a|} = \frac{|(u_x)_b(x_b - x_a) + (u_y)_b(y_b - y_a)|}{\sqrt{(x_b - x_a)^2 + (y_b - y_a)^2}} \tag{3.60}$$

where $X = (x, y)$, $U = (u_x, u_y)$.

These calculations should be performed for each node and for each neighboring node to find the minimum for the time step restriction. Since the velocities are zero on the boundaries, Δt_{cr}

Figure 3.5: Node sketch for stability calculations



should be not calculated for these nodes. Figure 3.5 gives an example of the nodes referred to in the equations above.

3.4 Equations of motion for a flexible plate

3.4.1 General equations with pressure and damping terms

The motion of a flexible plate can be defined by an equation, which is derived from Newton's Second Law:

$$\rho_m h\ddot{w} + d\dot{w} + B\nabla^4 w = -\delta p_m \tag{3.61}$$

where the flexural rigidity, B (Nm), is

$$B = \frac{Eh^3}{12(1-\nu^2)} \tag{3.62}$$

and

ρ_m is the surface density (kg/m^3),

h is the thickness (m),

w, is the displacement term (m),

$\dot{w}$, is the velocity term (m/s),

$\ddot{w}$ is the acceleration term(m/s^2),

d is the damping coefficient (kg/sm^2),

x is the axial location (m),

δp_m is the transmural surface pressure (N/m^2) $(\delta p_m = p_{upper} - p_{lower})$,

E is the elastic (Young's) modulus (N/m^2) and

ν is the Poisson ratio.

The equation of motion was discretized using an established finite difference method. There are advantages and disadvantages to an implicit method. The advantage is that stability can be maintained over much larger values of Δt than those used for an explicit approach. Hence, considerably fewer time steps are used for calculations over a given interval of t. This results in less computer time. As for the disadvantages, it is more complicated to set up and program. Also, since significant matrix manipulations are usually required at each time step, the computer time per time step is much larger than in the explicit approach. In addition, since large Δt can be taken, the truncation error is large, and the use of implicit methods to follow the exact time series (time variations of the independent variable) may not be as accurate as an explicit approach. (Note that the advantages and disadvantages of explicit schemes are given in Section 3.3.6.)

The finite difference form of Equation 3.61 at subsequent time steps for the i^{th} nodal displacement can be written as

$$\ddot{w}_i^t = \frac{1}{\rho_m h}(-d\dot{w}_i^t - B\nabla^4 w_i^t - \delta p_m^t) \tag{3.63}$$

$$\ddot{w}_i^{t+\delta t} = \frac{1}{\rho_m h}(-d\dot{w}_i^{t+\delta t} - B\nabla^4 w_i^{t+\delta t} - \delta p_m^{t+\delta t}) \tag{3.64}$$

Summing Equations 3.63 and 3.64 and rearranging for $\ddot{w}_i^{t+\delta t}$ we obtain

$$\begin{aligned} \ddot{w}_i^{t+\delta t} &= -\frac{B}{\rho_m h}(\nabla^4 w_i^t + \nabla^4 w_i^{t+\delta t}) \\ &\quad -\frac{d}{\rho_m h}(\dot{w}_i^t + \dot{w}_i^{t+\delta t}) - \ddot{w}_i^t - \delta p_m^t - \delta p_m^{t+\delta t} \end{aligned} \tag{3.65}$$

In Equation 3.65 the finite-difference operator at node i is given by

$$\nabla^4 w_i \equiv \left.\frac{\partial^4 w}{\partial x^4}\right|_i \equiv \frac{w_{i+2} - 4w_{i+1} + 6w_i - 4w_{i-1} + w_{i-2}}{\delta x^4} \tag{3.66}$$

The derivation of the finite-difference operator is shown in Appendix M.

In addition to Equation 3.61, the basis equations for the motion of any mass point of the discretized plate can be written in the form of

$$w_i^{t+\delta t} \approx w_i^t + \frac{\delta t}{2}(\dot{w}_i^t + \dot{w}_i^{t+\delta t}) \tag{3.67}$$

$$\dot{w}_i^{t+\delta t} \approx \dot{w}_i^t + \frac{\delta t}{2}(\ddot{w}_i^t + \ddot{w}_i^{t+\delta t}) \tag{3.68}$$

Therefore, the system of equations to be solved (in finite-difference form) to obtain the displacement, velocity and acceleration is defined by Equations 3.65, 3.67 and 3.68.

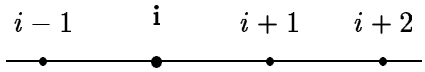
3.4.2 Equations of motion in vacuo without damping

The term $-\delta p_m$ is the pressure forcing term. In vacuo, where pressure forces do not exist, its value is set to zero. When damping is not accounted for, the damping coefficient d is also set to zero.

Under these conditions Equation 3.65 simplifies to

$$\ddot{w}_i^{t+\delta t} = -\frac{B}{\rho_m h}(\nabla^4 w_i^t + \nabla^4 w_i^{t+\delta t}) - \ddot{w}_i^t \tag{3.69}$$

Figure 3.6: Node numbering for the upstream end (i) of a flexible plate



3.4.3 End, initial and boundary conditions

End conditions

The boundary conditions, defining the conditions at the two ends of a flexible plate, are referred to as either hinged, clamped or free end conditions. Node numbering for the upstream end of a flexible plate is shown in Figure 3.6. Node $i-1$ represents a "dummy node" used in boundary condition calculations.

When the derivatives at the plate ends are set to zero (for example at $x = 0$), the conditions can be interpreted as the deflection ($w(0)$), the angle of inclination of the tangent of the deformed plate ($\partial w(0)/\partial x$), a quantity proportional to the bending moment ($\partial^2 w(0)/\partial x^2$) and a quantity proportional to the shear force ($\partial^3 w(0)/\partial x^3$). For arbitrary boundary conditions two of these conditions vanish.

For a *hinged–hinged* configuration the plate pivots around its leading and trailing edges. Therefore, the displacement and the bending moment are zero. For this, the end conditions are:

$$w_i = \frac{\partial^2 w_i}{\partial x^2} = 0 \tag{3.70}$$

since

$$\frac{\partial^2 w_i}{\partial x^2} = \frac{w_{i+1} - 2w_i + w_{i-1}}{\delta x^2} = 0 \tag{3.71}$$

the second derivative at $w_i = 0$ results in

$$w_{i-1} = -w_{i+1} \tag{3.72}$$

For a *clamped–clamped* configuration the first and last nodes are stationary. In addition, the tangent of the deformed rod between the end node and the first node inside the domain is zero (i.e., the 2^{nd} node and $N-1^{th}$ node have zero displacements). In mathematical terms the boundary condition can be described as

$$w_i = \frac{\partial w_i}{\partial x} = 0 \tag{3.73}$$

since

$$\frac{\partial w_i}{\partial x} = \frac{w_{i+1} - w_i}{\delta x} = 0 \tag{3.74}$$

the first derivative results in

$$w_{i+1} = w_i = 0 \tag{3.75}$$

For a *free end* configuration it is assumed that the bending moment and the shear force is zero at the last node. That is

$$\frac{\partial^2 w_i}{\partial x^2} = \frac{\partial^3 w_i}{\partial x^3} = 0 \tag{3.76}$$

Calculation of the finite difference operator using free end conditions requires the knowledge of the displacement of two additional "dummy" points outside of the physical nodes of the plate.

From Equation 3.71 the first dummy point can be calculated as

$$w_{i+1} = 2w_i - w_{i-1} \tag{3.77}$$

The third derivative is calculated from

$$\frac{\partial^3 w_i}{\partial x^3} = \frac{w_{i+2} - 3w_{i+1} + 3w_i - w_{i-1}}{\delta x^3} = 0 \tag{3.78}$$

Substituting Equation 3.77 to 3.78 gives the displacement value of the second dummy point

$$w_{i+2} = 3w_i - 2w_{i-1} \tag{3.79}$$

Therefore, displacement of the dummy nodes w_{i+1} and w_{i+2} are calculated from the known nodal displacement values of w_i and w_{i-1}.

3.4.4 Energies

For the cantilevered plate with unit width in the z direction the strain, E_s, kinetic, E_k, and total, E_t $(= E_s + E_k)$ energies are calculated at each time step of the simulation. The dimensional energy terms (in Nm) are calculated through the following expressions:

$$E_s = \frac{1}{2}B\int\left(\frac{\partial^2 w}{\partial x^2}\right)^2 dx \tag{3.80}$$

$$= \frac{1}{2}B\sum_{i=1}^{N}\left[\frac{w_{i+1}-2w_i+w_{i-1}}{\delta x^2}\right]^2 \delta x \tag{3.81}$$

$$E_k = \frac{1}{2}\rho_m h\int\left(\frac{\partial w}{\partial t}\right)^2 dx \tag{3.82}$$

$$= \frac{1}{2}\rho_m h\sum_{i=1}^{N}(\dot{w}_i)^2\delta x \tag{3.83}$$

The work done on the plate with unit width (W in Nm) is calculated as

$$W = \int\int\left(-\Delta p\frac{\partial w}{\partial t}dx\right)dt \tag{3.84}$$

$$= -\sum_{t_{start}}^{t_{end}}\left(\sum_{i=1}^{N}\Delta p_i\dot{w}_i\delta x\right)\delta t \tag{3.85}$$

3.4.5 Dimensionless variables

The data from the numerical experiments were post-processed using dimensionless variables. The time term was made dimensionless by the time required for a single cycle of oscillation:

$$\bar{t} = \frac{t}{t_\omega} \tag{3.86}$$

The total energy was normalized and non-dimensionalized by the total energy at the beginning of the time series $(t = 0)$ after a quasi-steady state condition is reached:

$$\bar{E}_t = \frac{E_t - E_0}{E_0} \tag{3.87}$$

The plate-restoring and pressure forces were non-dimensionalized by the fluid density and inlet velocity, using a ρU_{in}^2 scaling:

$$\bar{F}_{plate} = \frac{B\frac{\partial^4 w}{\partial x^4}}{\rho U_{in}^2} \tag{3.88}$$

and

$$\bar{\delta p} = \frac{\delta p}{\rho U_{in}^2} \tag{3.89}$$

3.5 Code coupling of the fluid and wall codes

This section outlines the methodology for the coupling of the fluid and wall codes. The coupled code can be divided into two main components. The first is virtually identical to the FEM Navier-Stokes code used in the validation exercise for duct flows with rigid walls. The velocity and pressure fields are initiated to zero and applying only the inlet velocity and no-slip boundary conditions, a quasi-steady-state condition is calculated, based on an RMS criterion. The only difference is in the domain geometry, where an initial displacement shape is prescribed to the cantilevered flexible section, which is held static during this phase of the calculations with an applied no-slip boundary condition.

Once a quasi-steady-state condition is reached, the time series starting time is re-set to zero. At this point the velocity and pressure fields are obtained, based on inlet and boundary conditions.

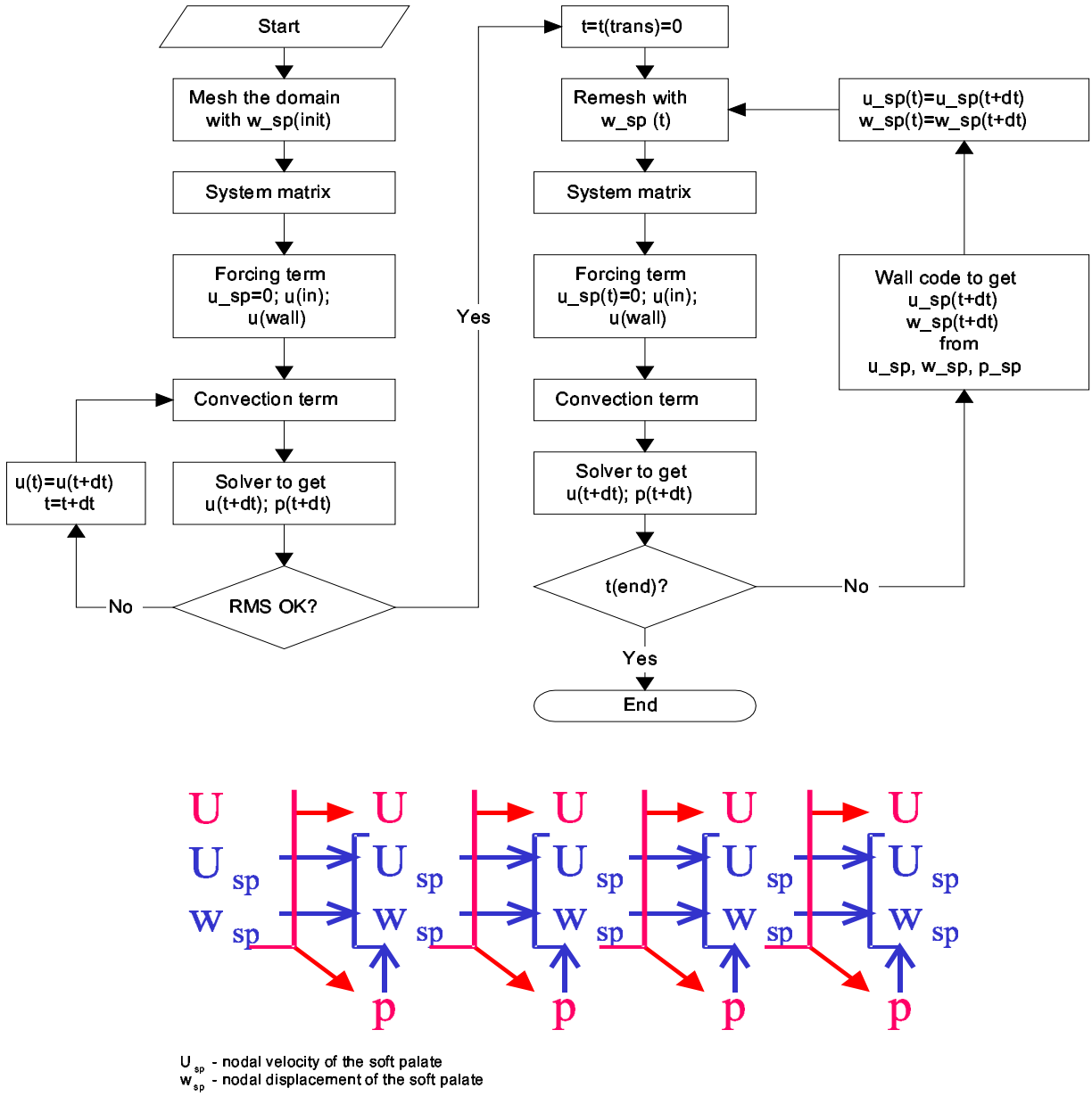
During code coupling it is assumed that the $U1$ nodal wall velocities – calculated in the wall code – are the same as the corresponding nodal velocities in the fluid code. The nodal displacements from the wall code are used to re-mesh the fluid domain. The acceleration term of the Navier-Stokes equations used in the FEM method is accounted for through $\frac{1}{\Delta t}M$, multiplied by the nodal velocities in the system of equations (see Equation 3.53). The nodal acceleration values of the wall code are calculated from the time derivative of the nodal velocity. Since the fluid code uses the same nodal velocity values for the cantilevered plate as the ones calculated

in the wall code, the nodal acceleration term will also match as long as an identical time step is used in both the fluid and wall codes.

The program logic used to calculate the fully coupled fluid–structure interaction time series is given in Figure 3.7. For each time step the domain is re-meshed, based on the wall displacement boundary conditions, which were obtained from the wall code. This results in a change of flow area for the affected elements, and introduces a systematic error, since the velocities are not adjusted at the displaced nodal locations. However, it is assumed that the incremental change in flow areas are very small and therefore can be neglected. For example, for the geometries used, the initial maximum compliant plate displacement is $w = 0.002$, compared to the half duct (one inlet channel) height of $h/2 = 0.005$. A single cycle of oscillation is modeled in 200 time steps, resulting in 50 time steps between zero and maximum displacements (that is for one quarter of a cycle). Therefore, one time step results in a 0.8% flow area change. For smaller initial displacements the change is even smaller. For $w = 0.001$ and $w = 0.0005$ initial maximum displacements the flow area change between time steps are 0.4% and 0.2 %, respectively.

Let us assume that at $t = 0$ (quasi-steady-state) the velocity field inside the domain is known. The cantilevered plate displacement is given from the wall code, while the nodal plate velocities are set to zero. From this, the fluid code can calculate the velocity and pressure fields at $t = t + \Delta t$. At this point the wall code is called, using the nodal velocities ($U1$), displacements (w) and transmural surface pressure ($p_{upper} - p_{lower}$) values across the plate, to calculate the new velocities and displacements at $t = t + \Delta t$. The new nodal displacement is used to re-mesh the domain, while the velocity values are used as boundary conditions in the fluid code. From this, the new velocity and pressure values can be calculated at $t = t + 2\Delta t$, starting the time iteration loop again. The time series stops when the time reaches the end time specified in the input file.

Figure 3.7: Full coupling of the fluid and wall codes



Chapter 4

Validation of the fluid code

This chapter documents a computer code validation exercise for the fluid codes, where the fluid code was validated against experimental and numerical data from open literature. This step was necessary to establish credibility for the codes before code coupling.

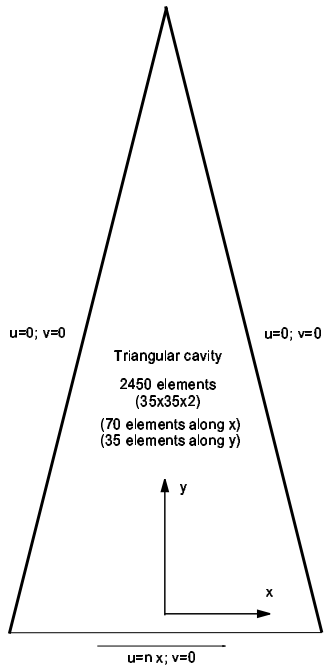
Code verification was completed by comparing results from MATLAB and FORTRAN codes (for the fluid code only). It was found that the two codes gave virtually identical results. In addition, for all computer programs the Debugger of the FORTRAN compiler was used to "step through" the codes, while monitoring key variables.

4.1 Lid-driven cavity flows

4.1.1 Lid-driven cavity flow for a triangular domain

A simple unit testing case is a lid-driven cavity flow in a triangular domain. The typical triangular domain with the sliding bottom lid boundary condition is shown in Figure 4.1. For the numerical test cases both the Stokes and the Navier-Stokes problems will be solved. Consider a triangular domain with two stationary rigid walls on the sides and the third side as the bottom lid, sliding at unit velocity from the left to the right.

Figure 4.1: Triangular cavity driven flow



Stokes flows

In the first stage of the code development, a prototype code was written in Matlab's scripting language to solve the Stokes equations. In the next stage the code was rewritten in FORTRAN. Even though the two codes are based on the same formulation, they can be considered semi-independent due to the differences between the two programming languages. For example, in the FORTRAN program memory allocation and deallocation was used to minimize memory usage. Different vector and matrix manipulations resulted in the re-formulation of the affected code sections. In addition, the second code used a sparse matrix solver by first sparsing the system matrix, then performing an LU decomposition and finally solving the system of equations using IMSL Mathematical Library functions for the last two items (functions DLFTXG and DLFSXG).

During the code development, partial results from the FORTRAN program were compared

against the corresponding Matlab generated results. It was found that the two calculations gave virtually identical results.

Moffatt [Mof64] presented some simple similarity solutions for the flow of a highly viscous fluid near a sharp corner between two planes on which a variety of boundary conditions were imposed. (Note that the singularities at the corners do not significantly affect the flow structure away from the corners.) The present numerical model recreates this sequence of increasingly weak vortices, created by a sliding lid.

Several runs were performed for various N_1 values. (N_1 represents the number of elements along the bottom of the domain, along the sliding lid.) The main characteristics of the domain for different runs are shown in Table 4.1, where N_1 is the number of element blocks along the bottom lid, N_2 is the number of element blocks along the y-axis, and h_X and h_Y are the element dimensions along the x- an y-axes, respectively. (Note that the FORTRAN code was run up to 2450 elements, while the Matlab code was tested up to 11250 elements. For reasons explained earlier (i.e., physical memory usage) it was decided that the FORTRAN code would not be tested above the 2450 element limit.)

The quality of the simulation can be also assessed by the number of vortices captured by the solution, while spatial convergence is measured by monitoring the location of the primary vortex, which is the one closest to the sliding lid.

A typical result for a 2450 element discretization is shown in Figure 4.2, where the (u_1) velocity component was plotted along the line of $x = 0$ from $y = 0$ to the top corner of the triangular domain at $y = +3.9575$. The upper figure shows the x-velocity components along the y-axis at $x = 0$, while the bottom figure indicates the number of captured vortices, for which the signum function is used to identify the zero crossings of the results. For a domain

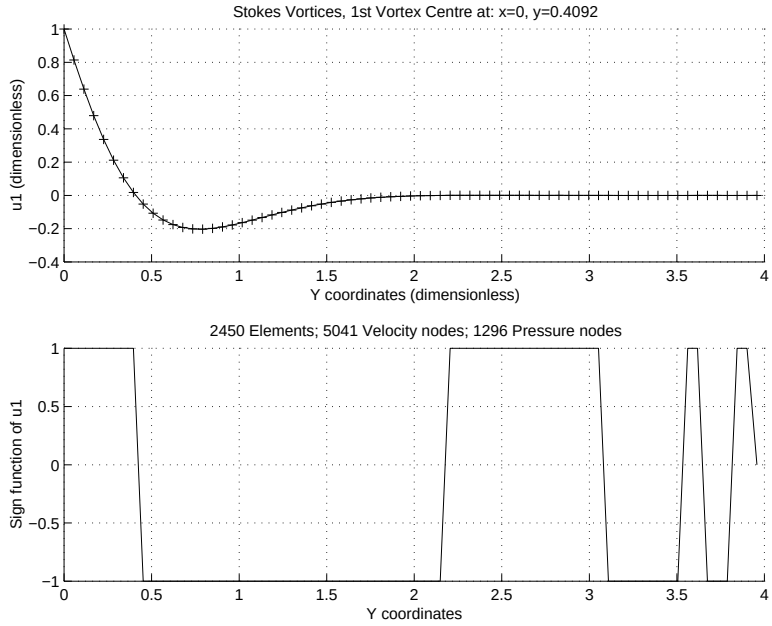
Table 4.1: Triangular domain discretization data for Stokes flow

N_1	N_2	h_X	h_Y	*Elements*	*Velocity nodes*	*Pressure nodes*
8	4	0.25	0.98938	32	81	25
10	5	0.2	0.7915	50	121	36
16	8	0.125	0.49469	128	289	81
20	10	0.1	0.39575	200	441	121
30	15	0.066667	0.26383	450	961	256
40	20	0.05	0.19788	800	1681	441
50	25	0.04	0.1583	1250	2601	676
70	35	0.028571	0.11307	2450	5041	1296
Matlab only						
100	*50*	*0.02*	*0.07915*	*5000*	*10201*	*2601*
150	*75*	*0.013333*	*0.052767*	*11250*	*22801*	*5776*

with 200 elements the solution captured 3 vortices. The captured vortices are shown in the bottom figure of Figure 4.2, by plotting the results as the signum function of u_1, the x-velocity component. (For each element of $u_1(X)$, $SIGN(u_1(X))$ returns 1 if the element is greater than zero, 0 if it equals zero and -1 if it is less than zero.) From the plots the captured vortices are identified by the zero crossings of the signum function. For the highest discretization the solution captured 6 vortices. However, in accordance with Moffatt [Mof64], the relative intensities of these additional vortices are several orders smaller than that of the first vortex. A comparison, however, between the results obtained using the signum function and streamtrace plots clearly show that the last few oscillations of the signum function are not physical. Figure 4.3 show a streamtrace plot for a 2450 element domain. While the signum function identified 5-6 sign changes based on velocity vector flow direction change, the streamtrace plot picked up only 3 vortex centers. Therefore, it is concluded that the sign changes close to the tip of the triangular domain are not physical and at the highest discretization with 2450 elements the code identified 3 vortices.

The dimensionless pressure field was also calculated for the domain. A typical pressure distri-

Figure 4.2: U1 velocities along the y-axis at x=0 [for a 2450-element domain]



bution for Stokes flow is shown in Figure 4.4. During the solution it was assumed the reference pressure node is at the upper tip of the domain, set to zero. Relative to this location there is a positive and a negative pressure region at the lower right and lover left corners, respectively. The positive and negative values are the result of the flow circulation direction inside the domain. Also, due to the nature of the flow, the pressure field pattern, just as the velocity field pattern is symmetrical.

Another way to present the results is by velocity distribution plots, shown in Figure 4.5, for only 200 elements since for a higher discretization the information presented would dominate the figure to a point, where it would become unreadable. The boundary condition at the bottom lid ($u = 1\hat{\mathbf{x}}$) dominates the velocity plot. These velocities are significantly larger than the subsequent velocities in the vortices. In addition, the relative vortex intensity dies down very quickly.

Figure 4.3: Streamtraces for a 2450 element domain

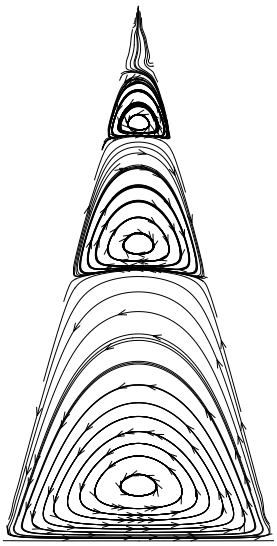
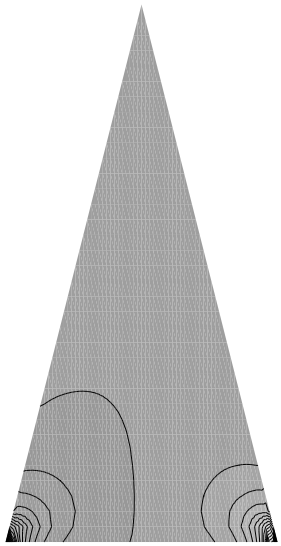


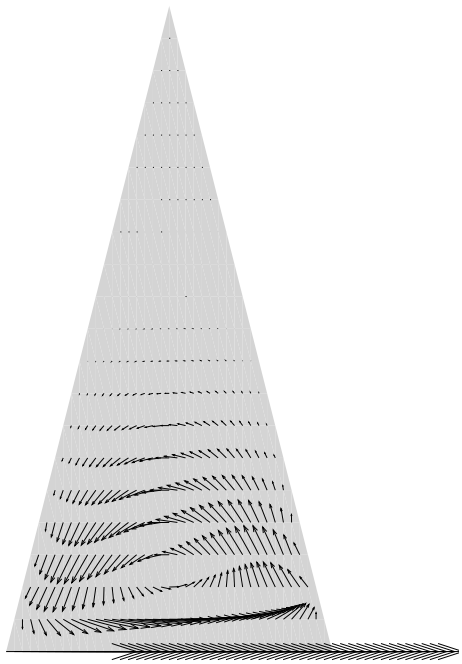
Figure 4.4: Pressure field for a 2450 element domain



In addition to the memory limitations, the execution time is affected by the size of the domain. While the execution time for the 32 element domain took only a few seconds, the execution time for the largest case with 75 elements along the y-axis and 150 elements at the bottom lid along the x-axis was 4032 seconds, or about 1 hours and 6 minutes, for the Matlab code. The FORTRAN code executed in 592 seconds for a 2450 element domain. The corresponding Matlab run required 391 seconds to execute. Although for the Stokes problems the Matlab code executed faster, it is expected that the FORTRAN code will out-perform it. It is due to the fact that the LU decomposition is performed only once in the FORTRAN code and subsequent iterations for the Navier-Stokes equations will result in much faster back substitution than the full matrix solution at each times step for the Matlab code.

Table 4.2 below provides the location of the 1st vortices for different number of elements in the domain. In the table N_1 is the number of element blocks along the bottom lid, N_2 is the

Figure 4.5: Velocity vectors for a 200-element domain



number of element blocks along the y-axis, and h_X and h_Y are the element dimensions along the x- an y-axes, respectively.

The location of the first vortex as a function of the element size (h_Y) is shown in Figure 4.6/A. (Figure 4.6/B shows a linear convergence of the results with the change in discretization The 45^o straight line represents ($dy/dx = 1$).) It can be seen from the figure and the table above that the location of the first vortex converges to 0.4 as $h \to 0$. Using this assumption, the error can be calculated between the calculated location of the first vortex and the "converged" value of 0.4, using the correlation:

$$Error_i = \frac{VC_i - 0.4}{0.4} \tag{4.1}$$

where $Error_i$ is the error for a domain with i number of elements, and VC_i is the 1st Vortex

Table 4.2: Location of first vortex centre from the bottom lid

N_1	N_2	*Elements*	h_X	h_Y	$1st\ Vortex$	$Error\ (fract)$
8	4	32	0.25	0.98938	0.62152	0.55381
10	5	50	0.2	0.7915	0.55477	0.38692
16	8	128	0.125	0.49469	0.47207	0.18017
20	10	200	0.1	0.39575	0.45452	0.1363
30	15	450	0.066667	0.26383	0.4326	0.081512
40	20	800	0.05	0.19788	0.4221	0.05525
50	25	1250	0.04	0.1583	0.4160	0.03998
60	30	1800	0.0333	0.13192	0.412	0.03
70	35	2450	0.028571	0.11307	0.40918	0.022947
Matlab only						
100	*50*	*5000*	*0.02*	*0.07915*	*0.40418*	*0.010451*
150	*75*	*11250*	*0.013333*	*0.052767*	*0.40036*	*0.00090443*

Centre for the corresponding domain. The results for different number of elements is shown in Table 4.2.

The convergence is shown in Figure 4.7, by plotting the calculated error against the elements size (h_Y) on a logarithmic scale. The slope of the resulting line for the first section (moving from larger to smaller element sizes) is approximately 1.57 (line of best fit), while for the last section it changed to 6, indicating a much faster convergence as $h \to 0$.

Another simple convergence check can be performed by comparing the change in element size with the change in location between different test runs. To achieve convergence, the change between the vortex location steps must be less than the change between the node size change. This is shown in Figure 4.6/B, which indicates a convergence of the runs.

As shown in Figure 4.2 (x velocity components along the y-axis at x=0) the first vortex gives velocities several orders of magnitude larger than those of subsequent vortices. This is in

Figure 4.6: Convergence of discretization based solutions

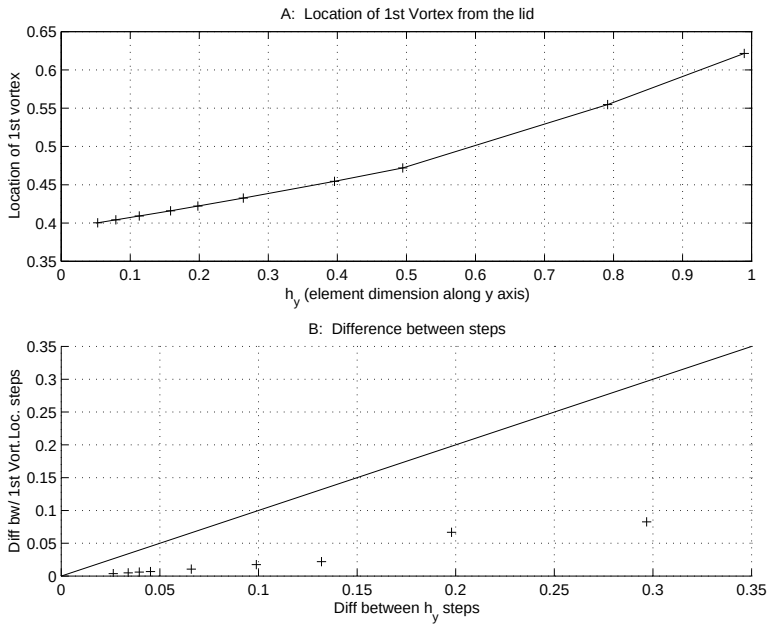


Figure 4.7: Convergence error at various discretizations

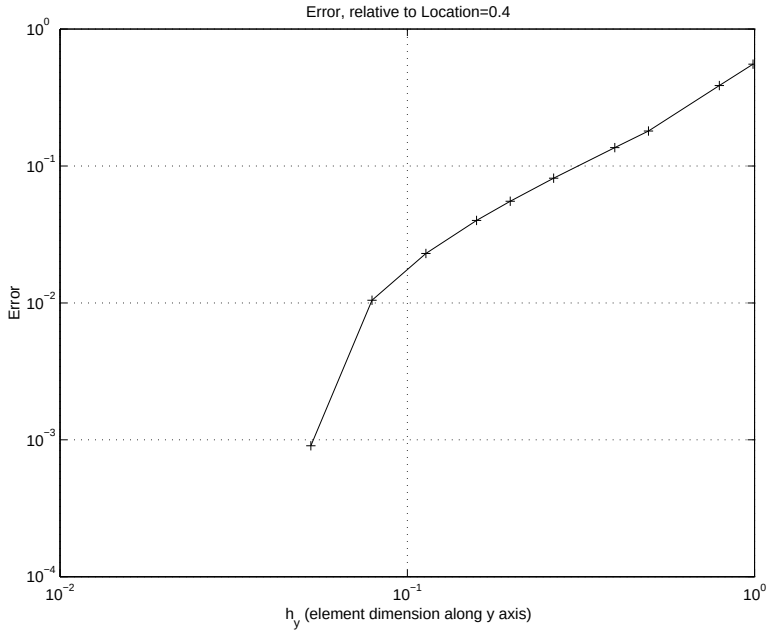


Table 4.3: Triangular domain discretization data for Navier–Stokes flow

N_1	N_2	h_X	h_Y	*Elements*	*Velocity nodes*	*Pressure nodes*
2	*1*	*1.0*	*3.9575*	*2*	*9*	*4*
4	*2*	*0.5*	*1.97875*	*8*	*25*	*9*
6	3	0.3333	1.31917	18	49	16
8	4	0.25	0.98938	32	81	25
10	5	0.2	0.7915	50	121	36
20	10	0.1	0.39575	200	441	121
30	15	0.0667	0.26383	450	961	256
40	20	0.05	0.19788	800	1681	441
50	25	0.04	0.1583	1250	2601	676
70	35	0.028571	0.11307	2450	5041	1296

agreement with Moffatt's findings, where he observed that the second vortex intensity was about 3 orders smaller than that for the first vortex, and the intensity for subsequent vortices were about 5 and 8 orders smaller than the intensity for the first vortex. The magnitude of the second and subsequent vortices are comparable to truncation errors in Taylor series. Therefore, it can be concluded that smaller vortices will not have a significant effect on parameters such as the total drag force on the lid. Also, the present numerical model recreated this sequence of increasingly weak vortices, created by a sliding lid.

Navier–Stokes problems

The main characteristics of the domain and the input variables for different runs based on the Navier–Stokes equations are shown in Table 4.3, where N_1 represents the number of elements along the bottom of the domain, along the sliding lid, while N_2 notes the number of elements along the y-axis at $x = 0$. *Elements* represents the number of elements in the domain and the number of velocity and pressure nodes are given at the end of the table.

Critical time step for the triangular domain

The stability of the calculations for the Navier-Stokes equations is achieved by the selection of an appropriate time step restriction, where the time step is less than the critical time step calculated with the method given in Section 3.3.6.

Due to the boundary conditions and the structured grid used for the wedge shaped domain (where the element distances are the same throughout the domain from element to element), the largest velocity given along the bottom lid will be the same (unchanged) for any time step for both Stokes or Navier-Stokes calculations. This results in the smallest critical time step by minimizing h'/u' (see Equation 3.58). (Note that the critical time step is applied to the Navier-Stokes calculations only. The Stokes calculations (without convection) are stable and they do not require stability restrictions.)

The flow field inside a two-dimensional wedge is driven by the sliding bottom lid. As shown in the numerical results for the Stokes conditions, the velocity decreases very rapidly with the distance from the bottom lid. Due to this rapid velocity drop the largest velocity occurs along the bottom lid, which is larger than the combined x and y velocity components anywhere inside the flow field. Therefore, the critical time step calculations are bounded by calculations performed at the bottom lid.

Equation 3.58 implies that the critical time step size can be calculated by maximizing (u') and minimizing (h'). As explained in the previous paragraphs, u' is maximized using the largest velocity at the bottom lid. h' would be minimized by finding a neighbouring node along the bottom lid.

The calculated critical time steps for different grid sizes are summarized in Table 4.4.

Table 4.4: Critical time step (Δt_{cr}) for the wedge

N_2	N_1	*Elements*	h_x	$h' = \frac{h_x}{2}$	$\frac{h'}{u'} \cdot 0.723$	Δt_{crit}
1	2	2	1	0.5	0.3615	0.3254
2	4	8	0.5	0.25	0.1808	0.1627
3	6	18	0.3333	0.1667	0.1205	0.1085
4	8	32	0.25	0.125	0.0904	0.0813
5	10	50	0.2	0.1	0.0723	0.0651
10	20	200	0.1	0.05	0.0362	0.0325
15	30	450	0.0667	0.0333	0.0241	0.0217
20	40	800	0.05	0.025	0.0181	0.0163
25	50	1250	0.04	0.02	0.0145	0.0130
30	60	1800	0.0333	0.01667	0.0121	0.0108
35	70	2450	0.0286	0.01429	0.0103	0.0093

The Navier-Stokes equations will be calculated with the critical time step values given in Table 4.4. Ideally, the critical time step should be recalculated between temporal iterations. However, in the calculations the boundary conditions are constant, and hence it is assumed that any change in critical time steps is accounted for by the safety factor ($SF = 0.9$).

Spatial convergence

During spatial convergence testing the number of elements in the domain is increased and the results are compared against each other. Spatial convergence is achieved when further refinement of the mesh does not produce significant increase in the precision of the calculated variables.

The numerical testing of the Navier-Stokes solver for a triangular domain is performed at a Reynolds-number of 100. This low Reynolds-number was chosen for several reasons. At lower Re numbers the flow is laminar. It produces slightly different results from the Stokes solution, namely the vortices are expected to move from the center line along with the moving bottom lid. However, these deviations are not prominent enough to result in irregular vortices. The

Table 4.5: Spatial convergence for a triangular domain

N_2	*Elements*	h_x	*1st vortex*	*CPU time(s)*
3	18	0.3333	0.6741	0.70
4	32	0.25	0.5564	1.16
5	50	0.2	0.493	1.98
10	200	0.1	0.3919	17.38
15	450	0.0667	0.3579	47.38
20	800	0.05	0.3503	135.92
25	1250	0.04	0.3461	357.42
30	1800	0.0333	0.3435	637.48
35	2450	0.0286	0.3418	926.17

time step (Δt) was fixed at 0.01, while the number of iterations was set to 450. As the time step and the number of iterations were kept constants, the results for various nodalizations could be compared under the same time series conditions. The geometries used for the test runs are shown in Table 4.5.

Just as shown for the Stokes problem, the sliding bottom lid resulted in a circulating internal flow. The predictions showed the increasingly weaker vortices inside the domain. A good comparison of the results and determining convergence can be the location of the first vortex centre from the bottom lid. For the Stokes vortices the vortex centers aligned along the centre axis of the domain. As mentioned above, the Navier-Stokes vortices do not align along the same path. However, a comparison of the flow direction change along the same y-axis (at $x = 0$) can be used for the comparison. The location of the flow direction change is given in Table 4.5. The results are also plotted against the element size in Figure 4.8. It can be seen that with the refinement of the mesh the location of the first flow direction change converges to a value close to the location predicted for the 2450 element discretization.

Figures 4.9 and 4.10 show the RMS change of the velocity results between iteration steps.

Figure 4.9 contains the first 4 (coarse) discretizations. The best results obtained with the highest refinement (2450 elements) is plotted on the figure for comparison. For 18 elements the calculations did not converge. With further refinements the runs produced increasingly convergent results, and the results for a 200 element domain followed the expected RMS change trend, while not reaching the final convergence criteria. Figure 4.10 gives the results for 450, 800, 1250, 1800 and 2450 elements, respectively. It can be seen that these solutions reached the convergence criteria by the last iteration step. While all of these results would reach a convergent solution, the run with the highest discretization would have the highest convergence rate. It should be noted that with the reduction of the elements the time step can be increased according to the stability criteria. While that would reduce the number of iterations, the introduced errors would be larger. This issue will be examined in the next section for temporal convergence.

It was concluded that a discretization with 2450 elements would provide reasonably refined results, while keeping the computational time and resources low.

Temporal convergence

During temporal convergence testing the time step (Δt) is varied, while the discretization and the flow conditions are kept constant. The results are compared against each other. Temporal convergence is achieved when further reduction in the time step does not produce significant increase in the precision of the calculated variables.

The numerical testing of the Navier-Stokes solver for a triangular domain is performed at a Reynolds-number of 100. The number of iterations was set to 450, for a domain with 1250 elements. For this configuration the code produced converged results in 391 iteration steps using a time step of 0.016. This is used as a benchmark and from it the time series time was set to 5.1 for all of the temporal convergence test cases. The time step is varied from the stable to the unstable regions as shown in Table 4.6.

Figure 4.8: Flow direction change in a wedge

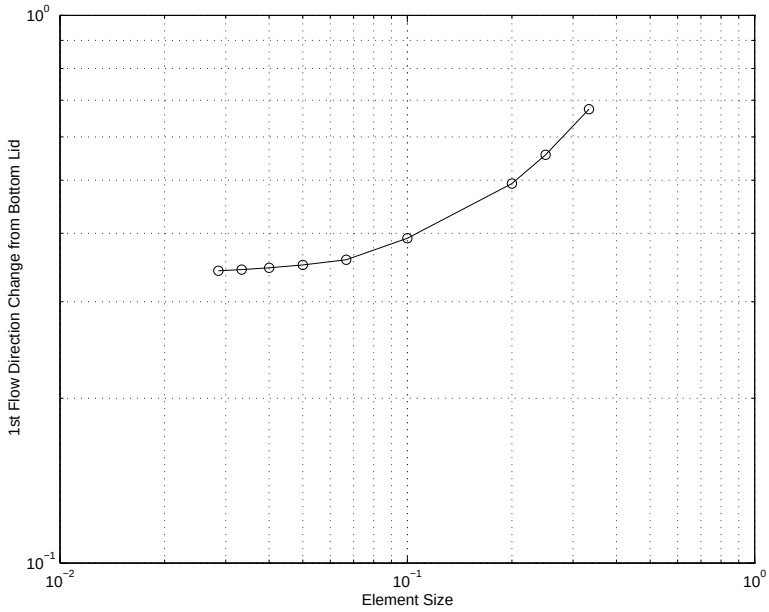


Figure 4.9: Spatial convergence in a wedge - RMS changes (a)

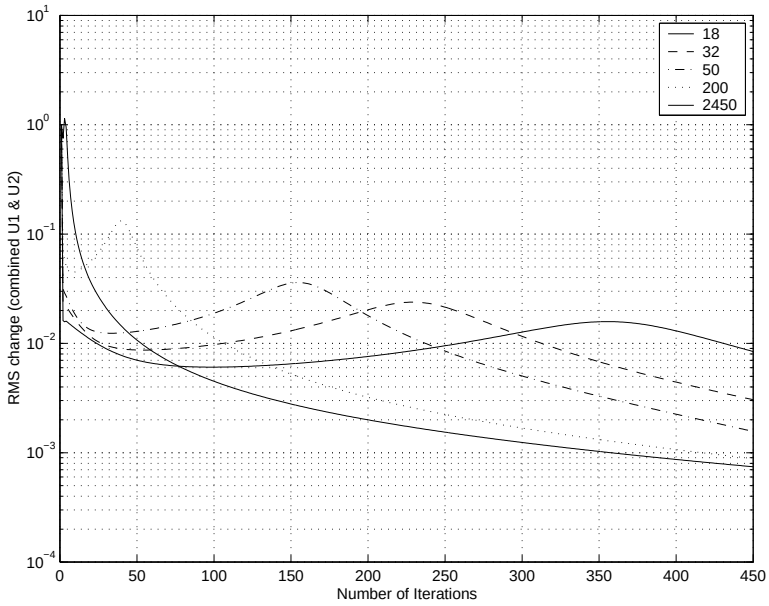
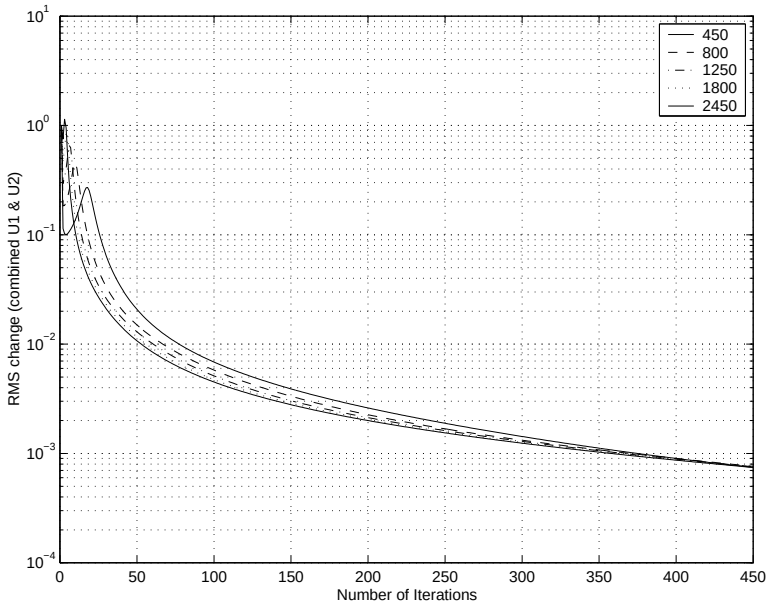


Figure 4.10: Spatial convergence in a wedge - RMS changes (b)



The results for the temporal convergence tests are shown in Figures 4.11 and 4.12. It can be seen that the code is stable for time steps below the stability criteria and for a number of values above it, up to $\Delta t = 0.075$. It is almost 6 times higher than the one calculated with Equation 3.58. When the safety factor is not used, it drops back to about the factor of 5, which is still a significant difference. This "discrepancy" can be easily explained. The critical time step was calculated from the smallest gap and the highest velocity. The highest velocity was conservatively equated with the bottom lid velocity. However, as shown by Moffatt [Mof64] and through the previous numerical test for Stokes flow, the flow velocity inside the domain is significantly lower than the bottom lid velocity, which would results in a higher Δt_{crit} than used here.

As expected, smaller time steps result in faster convergence rates. In the stable region the RMS changes follow very similar trends for all predictions. It was found that there are no

Table 4.6: Temporal convergence for a triangular domain

Δt	*Iterations*	$t_{time\ series}$	*CPU time*
0.002	2550	5.1	929.82
0.005	1020	5.1	453.72
0.01	510	5.1	379.42
0.013	**392**	5.096	309.61
0.025	204	5.1	272.97
0.05	102	5.1	294.38
0.075	68	5.1	280.45
0.1	51	5.1	262.81
0.255	20	5.1	213.87
0.51	10	5.1	213.64
1	5	5	217.7

significant changes in the solutions at the same time series time using different time steps in the stable region. For example, the difference between velocity predictions at 2500 iterations (using $\Delta t = 0.002$) and at 392 iterations (using $\Delta t_{crit} = 0.013$) is less than 0.4% for 95% of the non-zero velocity values. Runs performed at the lower time step range are considered more accurate, since at lower time steps the introduced accumulated numerical errors are smaller, when an explicit numerical scheme is used, such as in this case.

Numerical results for the triangular domain

The Navier-Stokes code was tested through a wide range of conditions to check the stability of the code, and to test the validity of the predictions. The numerical test matrix is shown in Table 4.7. The domain discretization was varied between 50 and 2450 elements, while the Reynolds-number was changed between 1 and 2000. The time step was set according to the critical time step criteria (Eq.3.58).

The convergence criterion was set to 1×10^{-3}. The number of iterations to reach this convergence criterion is also shown in Table 4.7.

Figure 4.11: Temporal convergence in a wedge - RMS changes (a)

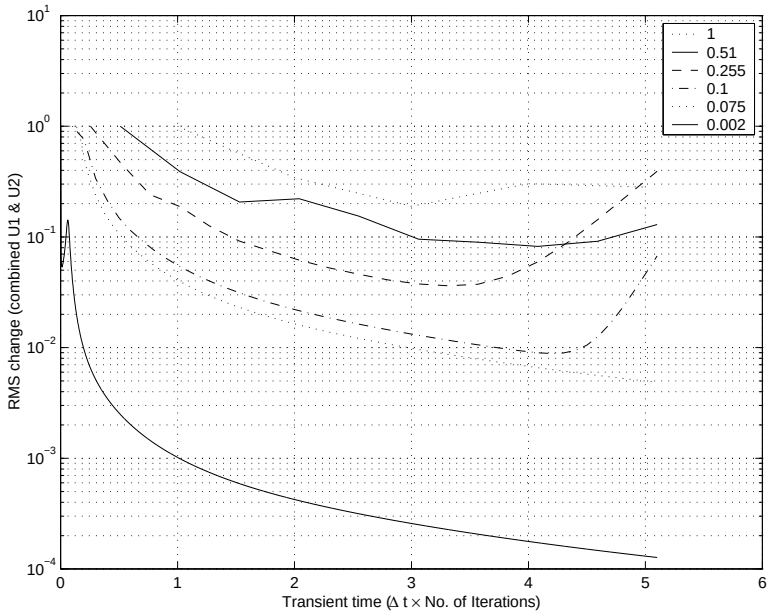


Figure 4.12: Temporal convergence in a wedge - RMS changes (b)

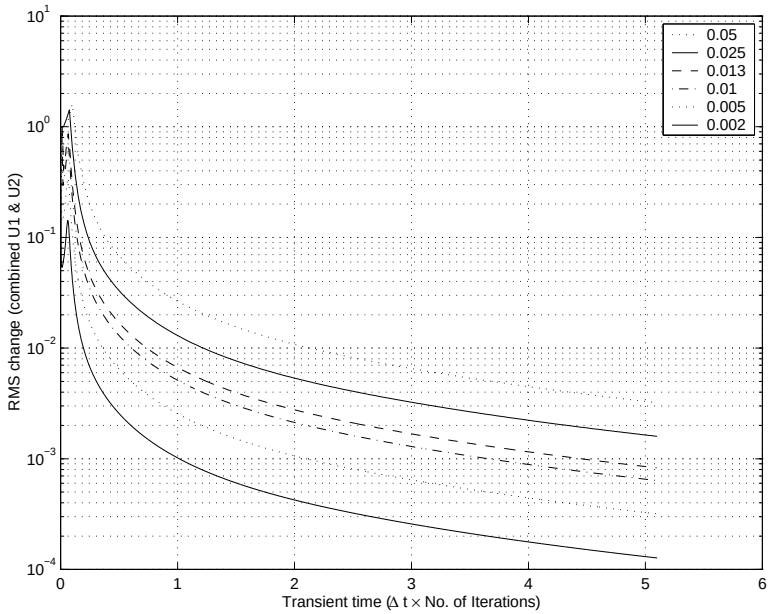


Table 4.7: Number of iterations to reach convergence for a wedge

No. of Elem.	50	200	800	1250	1800	2450
N_2	5	10	20	25	30	35
Δt	0.065	0.033	0.016	0.013	0.011	0.0093
Re						
1	9	13	20	23	25	29
50	106	179	254	277	296	315
100	131	264	361	391	413	435
250	256	463	620	657	692	714
500	501	352	926	966	1007	1046
1000	1275	1180	1509	1330	1344	1359
1500	1689	1491	1139	1771	1682	1602
2000	*500**	*3000**	2762	2550	1799	1876
$^*\Delta t = 0.01$	**No Conv.*	**No Conv.*				

The time series run times are calculated from the time step and the number of iterations. This number indicates when the run met the convergence criterion. It is assumed that when the convergence criterion is reached, the velocity change between iteration steps is very small, and the flow inside the domain can be considered "quasi steady-state". The time series times are shown in Table 4.8.

As an indication of the code's performance on the computer used for the numerical investigations, the CPU time for each run is shown in Table 4.9. It can be seen from the number of iterations (Table 4.7) and the CPU times (Table 4.9) that for a significant increase in iteration numbers – using the same domain discretization – the CPU time does not increase proportionately. It was found that the largest computational effort is dedicated to the LU decomposition of the system matrix. The actual iteration steps required only back substitution, and therefore, minimal computational effort and CPU time.

Table 4.8: Time series time at convergence for a wedge

No. of Elem.	50	200	800	1250	1800	2450
N_2	5	10	20	25	30	35
Δt	0.065	0.033	0.016	0.013	0.011	0.0093
Re						
1	0.585	0.429	0.32	0.299	0.275	0.2697
50	6.89	5.907	4.064	3.601	3.256	2.9295
100	8.515	8.712	5.776	5.083	4.543	4.0455
250	16.64	15.279	9.92	8.541	7.612	6.6402
500	32.565	11.616	14.816	12.558	11.077	9.7278
1000	82.875	38.94	24.144	17.29	14.784	12.6387
1500	109.785	49.203	18.224	23.023	18.502	14.8986
2000	*No Conv.*	*No Conv.*	44.192	33.15	19.789	17.4468

Table 4.9: CPU time at convergence for a wedge

No. of Elem.	50	200	800	1250	1800	2450
N_2	5	10	20	25	30	35
Δt	0.065	0.033	0.016	0.013	0.011	0.0093
Re						
1	$0.2s$	$3.1s$	$87.6s$	$276.9s$	$715.3s$	$835.8s$
50	$0.5s$	$7.0s$	$99.4s$	$288.3s$	$487.3s$	$945.6s$
100	$0.6s$	$10.4s$	$161.3s$	$306.3s$	$748.1s$	$1199.4s$
250	$1.2s$	$15.1s$	$161.5s$	$348.0s$	$713.1s$	$1042.9s$
500	$2.1s$	$12.4s$	$207.8s$	$544.8s$	$970.0s$	$1216.2s$
1000	$5.1s$	$37.0s$	$358.4s$	$827.4s$	$1523.4s$	$1778.8s$
1500	$6.6s$	$46.1s$	$357.0s$	$880.1s$	$1450.3s$	$2276.9s$
2000	*No Conv.*	*No Conv.*	$628.0s$	$12325.1s$	$1485.4s$	$2972.9s$

Velocity distribution results for the Navier–Stokes flows

Due to the size limitation of the thesis, only the results for the highest discretization are given here. At lower resolutions the results show similar trends. However, some of the flow features are not fully resolved. For the 2450 element domain, there are 35 elements along the y-axis and 70 elements along the x-axis. This translates to 71 and 141 velocity nodes along the y- and x-axes, respectively. This is the highest domain resolution examined in the present analysis. While the code has no limitations on increasing the mesh size even higher, the computing resources and the execution time limited further mesh refinements. However, it was found that a domain with 2450 elements captures the flow conditions sufficiently.

The code captured 4 vortices at Re=1 (see Figure 4.13). However, at this discretization the third vortex remained captured up to Re=2000 (see Figure 4.20). At this Reynolds-number the flow conditions at the right side wall were slightly disturbed, but the code provided consistent results for the entire Reynolds-number range examined.

The migration of the vortices upward seems physical. With the decrease of the fluid viscosity the Reynolds-number increased. At lower viscosity the influence from one vortex to another is much weaker. This explains the reduction of the number of vortices. It should be noted that the intensity of these subsequent vortices are several magnitude less than that for the first vortex, close to the bottom lid. This is not obvious from the streamtrace plots, where the vortex intensity is not shown. However, due to this magnitude difference, vector plots would also fail to properly visualize the flow field.

Pressure results for 2450 elements

Besides the flow conditions the code also calculated pressure values at the pressure nodes. For a 2450 element domain the pressure was calculated at 1296 nodal points. The calculated pressure values are dimensionless, and they are calculated relative to a reference node set at

Figure 4.13: Streamtraces for 2450 El.s (1)

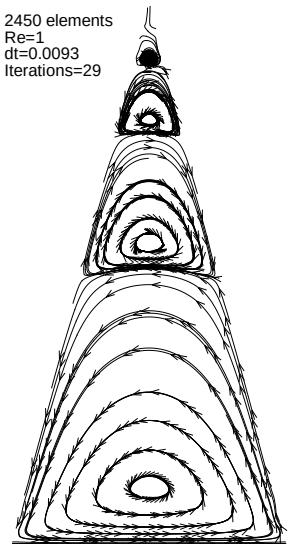


Figure 4.14: Streamtraces for 2450 El.s (2)

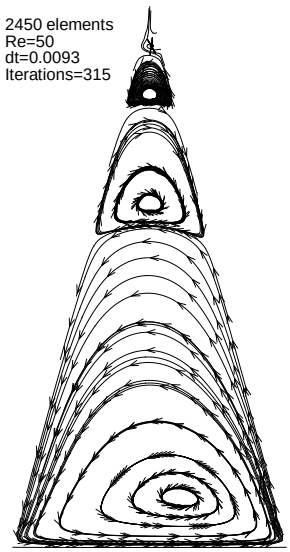


Figure 4.15: Streamtraces for 2450 El.s (3)

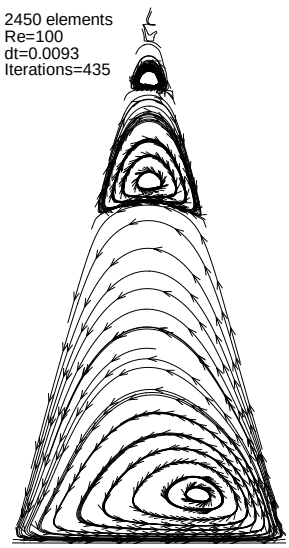


Figure 4.16: Streamtraces for 2450 El.s (4)

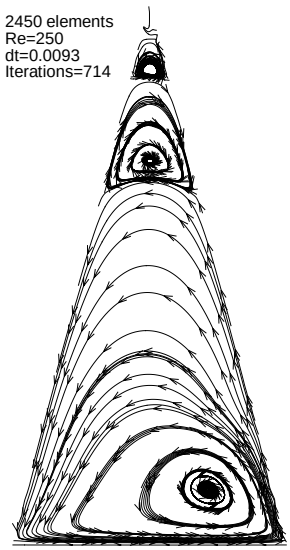


Figure 4.17: Streamtraces for 2450 El.s (5)

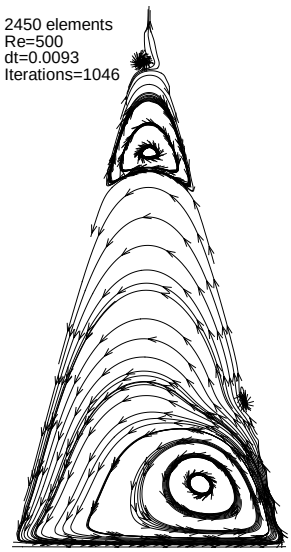


Figure 4.18: Streamtraces for 2450 El.s (6)

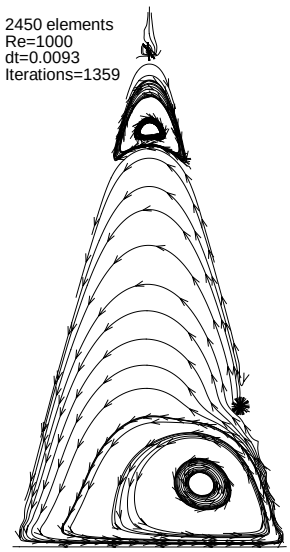


Figure 4.19: Streamtraces for 2450 El.s (7)

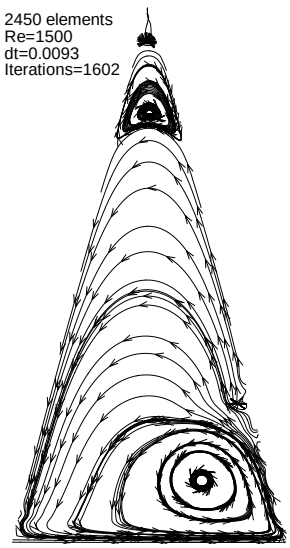


Figure 4.20: Streamtraces for 2450 El.s (8)

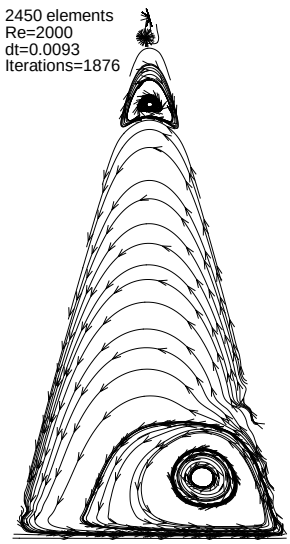


Figure 4.21: Pressure at Re=1

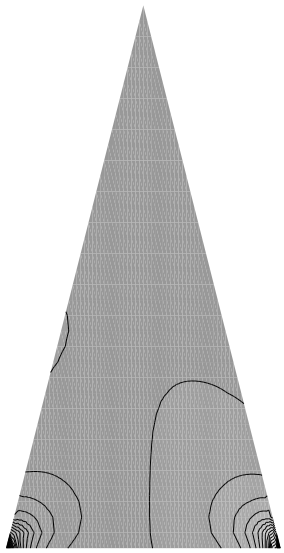
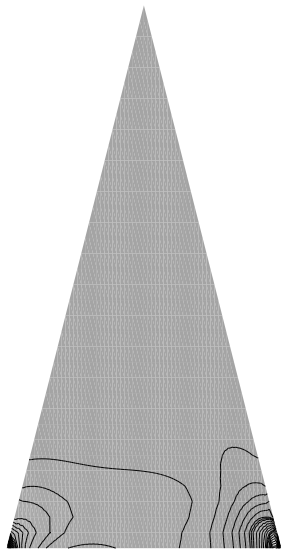


Figure 4.22: Pressure at Re=50



the top tip of the domain. At this reference point the pressure is set to zero, and all other points are calculated relative to it.

At Re=1 (see Figure 4.21) the pressure contours are very similar to those of the Stokes pressure contours. Due to the flow conditions, the pressure is negative at the left bottom corner and positive at the right bottom corner. This is consistent with the circulation pattern of the fluid along the bottom lid. With the increase of the Reynolds-numbers, this balance between the corners shifted towards the right hand corner. As the center of the primary vortex moved along the positive x-axis, and the local flow conditions were amplified at that corner, the local pressure at the right hand corner became significantly larger than that in the left hand corner. For example, at Re=1 the pressure values at the left and right hand corners were -68 and +63, respectively. The corresponding values at Re=2000 (see Figure 4.28) were -0.05 and 0.35. Note that these pressure values, just as the calculated velocity values, are dimensionless.

Figure 4.23: Pressure at Re=100

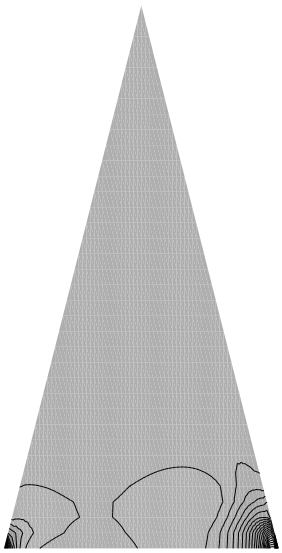


Figure 4.24: Pressure at Re=250

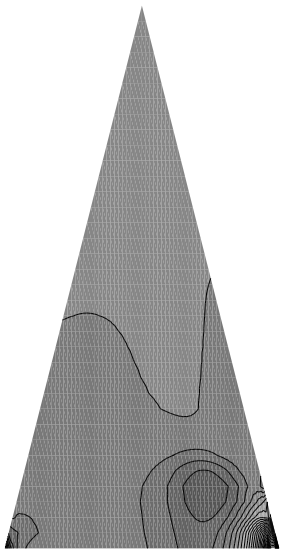


Figure 4.25: Pressure at Re=500

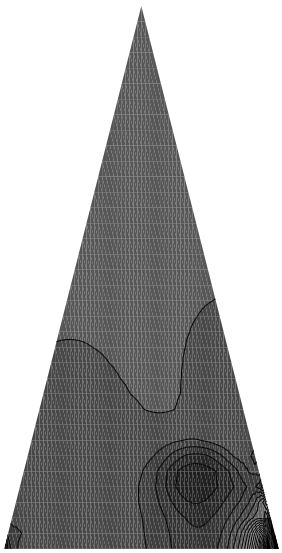


Figure 4.26: Pressure at Re=1000

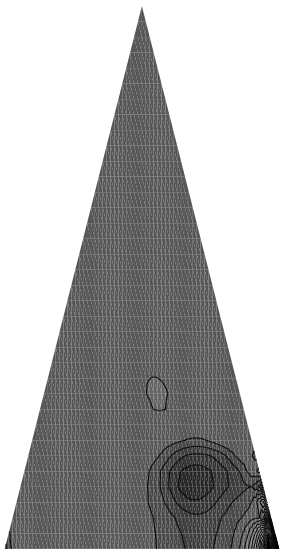


Figure 4.27: Pressure at Re=1500

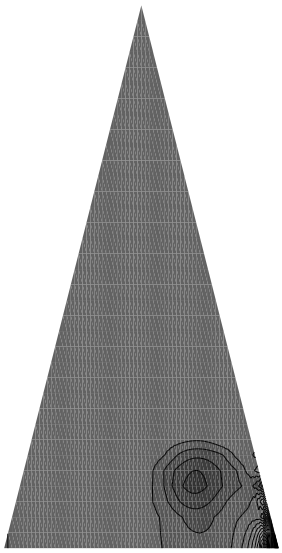
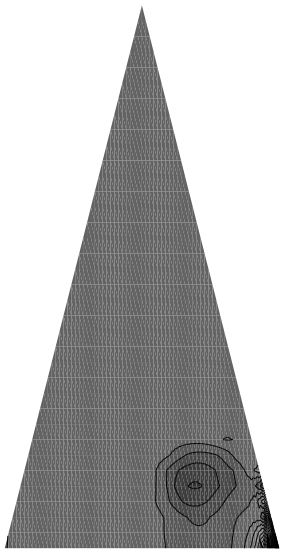


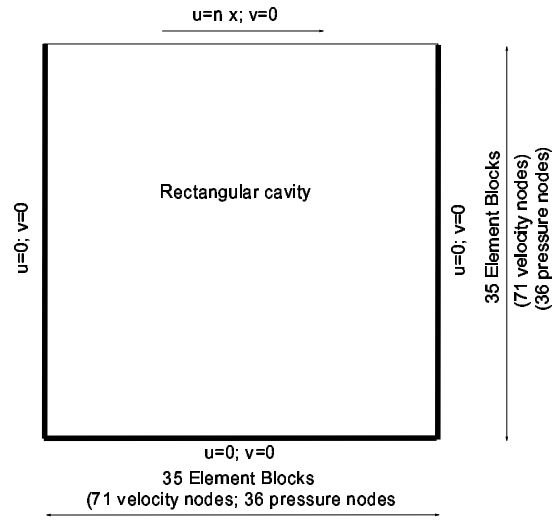
Figure 4.28: Pressure at Re=2000



4.1.2 Lid-driven cavity flow for a square domain

Consider the laminar flow of a viscous, incompressible fluid in a square cavity bounded by three motionless walls and a lid moving at a constant velocity in its own plane. The third dimension is assumed to be long enough to have a plane flow. Along all walls except the top one, the velocity is required to vanish. Along the top wall the normal velocity component vanishes and the tangential components are prescribed constants. The Reynolds-number for the flow, for a fluid of given dynamic viscosity, is determined by the size of the box and the magnitude of the non-vanishing velocity component of the boundary. Figure 4.29 illustrates the basic geometry used. Previous works using similar geometries were documented by Pan & Acrivos [PA67], Cheng [Che72], Kawahara *et.al.* [et.76] and by Reddy & Gartling [RG94]. Singularities exists at each corner where the moving lid meets a fixed wall. In spite of these difficulties, the driven cavity problem is too easy a test for algorithms for the numerical simulation of viscous incompressible flows. The reason for this is that it is rather easy to generate solutions that

Figure 4.29: Rectangular cavity driven flow



have the global features one would expect in such a flow. Furthermore, since the problem is for the most part not physically realizable, one cannot compare the numerical solution with meaningful experimental data.

Several runs were performed for various discretizations. The main characteristics of the domain and the input variables for different runs are shown in Table 4.10, where N_x represents the number of elements along the x-axial, along the sliding lid, while N_y notes the number of elements along the y-axis at $x = 0$. It should be noted that there are the same number of element blocks along the x-axis as along the y-axis, due to the domain geometry. *Elements* represents the number of elements in the domain and the number of velocity and pressure nodes are given at the end of the table.

Stokes problem

The Stokes problem can be viewed as a simplified version of the Navier-Stokes equations, where the convection and time dependent terms drop out and fluid is considered highly viscous (see

Table 4.10: Lid-driven square cavity domain discretization data

$N_x = N_y$	$h_X = h_Y$	*Elements*	*Velocity nodes*	*Pressure nodes*
5	0.2	50	121	36
8	0.125	128	289	81
10	0.1	200	441	121
12	0.083	288	625	169
14	0.071	392	841	225
20	0.05	800	1681	441
30	0.033	1800	3721	961
35	0.029	2450	5041	1296

Table 4.11: CPU times for the Stokes lid-driven square cavity flow runs

$N_x = N_y$	$h_X = h_Y$	*Elements*	*Velocity nodes*	*Pressure nodes*	*CPU time*
5	0.2	50	121	36	$0.14s$
10	0.1	200	441	121	$1.52s$
14	0.071	392	841	225	$6.22s$
20	0.05	800	1681	441	$44.14s$
30	0.033	1800	3721	961	$352.73s$
35	0.029	2450	5041	1296	$643.30s$

Equation 3.2).

Several runs were performed for various discretizations. The main characteristics of the domain for different runs are shown in Table 4.11. For reasons explained earlier (i.e., physical memory usage) it was decided that the FORTRAN code will not be tested above the 2450 element limit.

The results are presented as velocity values along the x and y geometric centers. In addition, a streamtrace plot show the captured vortices in the domain for a 2450 element domain.

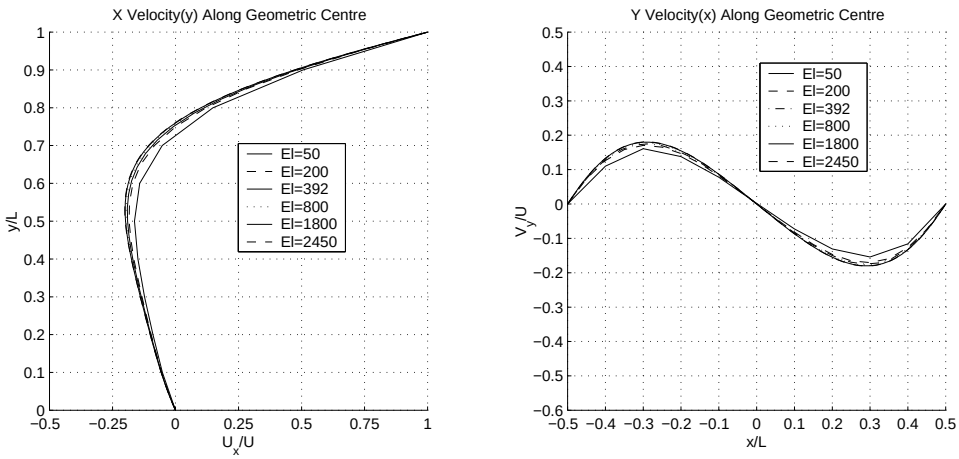
Reddy & Gartling [RG94] presented solutions for various mesh sizes and at various Reynolds-

numbers. These will be compared with the results in the section examining conditions for the Navier-Stokes problem (see Section 4.1.2). However, at Re=1 the results can be compared with Stokes results. It was found that at Re=1 the result were in very good agreement with Reddy's results.

The results for the runs representing the Stokes problem are shown in Figure 4.30. The figure on the left hand side show the x velocity components along the y geometric axis of the domain. The velocity at the top is equal to the velocity of the sliding top lid (the top lid was set to move at unit velocity, that is $u_{lid} = 1$). Along the bottom wall the velocity is zero, due to the no slip boundary condition. The flow circulation inside the domain resulted in a symmetrical flow pattern, where the center of the primary vortex aligned with the geometric axis. The figure on the right hand side show the y velocity components along the x geometric axis of the domain. The results show symmetrical flow patterns in opposite directions. With the increase of the number of elements, the results are more and more refined. However, with the increase of the number of elements in the domain, the size of the system matrix increases significantly. Therefore, the size of the domain is limited by the memory limitations of the computer. Note that the FORTRAN program used sparse matrices, which significantly reduced the size of the system matrix. Information on the non-zero elements in the system matrix is given in Section E.3.3.

The quality of the simulation can be also assessed by the number of vortices captured by the solution. These can be shown in a streamtrace plots for 200 and 2450 elements (see Figures 4.31 and 4.32). Between 50 and 392 elements only the primary vortex is captured. At 800 elements the first secondary vortex is captured at the bottom right hand corner. Discretizations for 1800 and 2450 elements capture two secondary vortices, in the left and right bottom corners of the cavity. These flow pattern results are consistent with the expected symmetrical flow patterns for Stokes flow.

Figure 4.30: Velocities at the geometric centers for Stokes flow



The primary vortex has velocities which are several orders of magnitude larger than those for the corner vortices. This is in agreement with Moffatt's [Mof64] findings, where – for a triangular domain – he observed that the intensity of the secondary vortex was about 3 order smaller than that for the primary vortex.

Navier-Stokes problem

Critical time step for the square cavity domain

The stability of the calculations for the Navier-Stokes equations is achieved by the selection of an appropriate time step restriction, where the time step is less than the critical time step calculated with the method given in Section 3.3.6.

The flow field inside a two-dimensional cavity is driven by the sliding top lid. As shown in the numerical results for Stokes flow, the velocity decreases very rapidly with the distance from the top lid. Due to the boundary conditions and the structured grid used for the square cavity domain (where the element distances are the same throughout the domain from element to

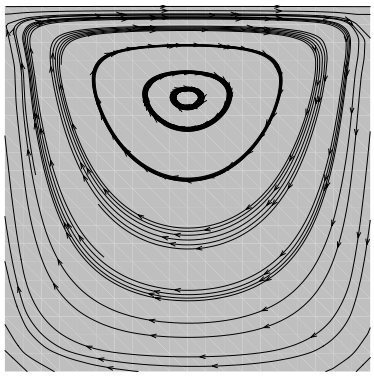


Figure 4.31: Stokes Streamtraces for 200 elements

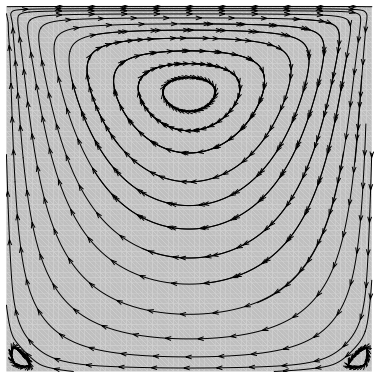


Figure 4.32: Stokes Streamtraces for 2450 elements

element), the largest velocity given along the sliding top lid will be the same (unchanged) for any time step for both Stokes or Navier-Stokes calculations. None of the combined x and y velocity components inside the flow field amount for local velocities above the top lid velocity. This results in the smallest critical time step by minimizing h'/u' (see Equation 3.58). (Note that the critical time step is applied to the Navier-Stokes calculations only. The Stokes calculations (without convection) are stable and they do not require stability restrictions.)

Equation 3.58 implies that the critical time step size can be calculated by maximizing (u') and minimizing (h'). As explained in the previous paragraphs, u' is maximized using the largest velocity at the bottom lid. h' would be minimized by finding a neighbouring node along the bottom lid.

The calculated critical time steps for different grid sizes are summarized in Table 4.12.

As mentioned in Section 3.3.6, there is a linear correlation between the element size and the critical time step. The results given in the above table confirm this. The ratio of both the

Table 4.12: Critical time step (Δt_{cr}) for the cavity

					Δt_{crit}		
$N_x = N_y$	*Elements*	h_x	$h' = \frac{h_x}{2}$	$\frac{h'}{u'(=1)} \cdot 0.723$	$for\ u = 1$	$for\ u = 0.5$	$for\ u = 0.33$
5	50	0.200	0.100	0.072	0.065	0.130	0.197
8	128	0.125	0.063	0.045	0.041	0.081	0.123
10	200	0.100	0.050	0.036	0.033	0.065	0.099
12	288	0.083	0.042	0.030	0.027	0.054	0.082
14	392	0.071	0.036	0.026	0.023	0.046	0.070
20	800	0.050	0.025	0.018	0.016	0.033	0.049
30	1800	0.033	0.017	0.012	0.011	0.022	0.033
35	2450	0.029	0.014	0.010	0.009	0.019	0.028

change in element size (h_x) and the change in critical time step (Δt_{cr}) is about 7.

The recommended critical time step values for the lid-driven square cavity flow using the Navier-Stokes equations are given in Table 4.12. Ideally, the critical time step should be recalculated between temporal iterations. However, in the calculations the boundary conditions are constant, and hence it is assumed that any change in critical time step is accounted for by the safety factor ($SF \approx 0.9$). (Note that in the calculations the safety factor was reduced to as much as $SF = 0.45$ for further stability.)

Spatial convergence

During spatial convergence testing the number of elements in the domain is increased and the results are compared against each other. Spatial convergence is achieved when further refinement of the mesh does not produce a significant increase in the precision of the calculated variables.

The numerical testing of the Navier-Stokes solver for a square domain is performed at a Reynolds-number of 400. This low Reynolds-number was chosen for several reasons. At lower

Table 4.13: Spatial convergence for a square domain

N_x	*Elem.*	U_{lid}	*Re*	Δt_{crit}	Δt_{used}	*Conv.Crit.*	*Iterations*	*CPUtime*
5	50	0.5	400	0.13	0.07	$5.00E-05$	593	2.89
8	128	0.5	400	0.081	0.04	$5.00E-05$	864	18.56
10	200	0.5	400	0.065	0.033	$5.00E-05$	986	37.18
12	288	0.5	400	0.054	0.027	$5.00E-05$	1141	67.3
14	392	0.5	400	0.046	0.023	$5.00E-05$	1282	129.22
20	800	0.5	400	0.032	0.016	$5.00E-05$	1670	389.12
30	1800	0.5	400	0.022	0.011	$5.00E-05$	2183	1271.17
35	2450	0.5	400	0.018	0.009	$5.00E-05$	2516	2129.42

Re numbers the flow is laminar. It produces slightly different results from the Stokes solution, namely the vortices are expected to move from the center line along with the moving top lid. Also, these conditions match the conditions by Reddy & Gartling [RG94], allowing for direct comparison of the results. The time step was set according to the stability criterion, while the number of iterations was determined by the convergence criterion of 5×10^{-5}. The geometries, initial conditions and the iteration results for the test runs, are shown in Table 4.13.

Just as shown for the Stokes problem, the sliding top lid resulted in a circulating internal flow. The predictions showed the development of the primary vortex in the middle of the domain and the weaker secondary vortices at the lower left and right corners inside the domain. The normalized x and y velocity components for the runs are given in Figures 4.33 and 4.34 for various discretizations. It can be seen that with the refinement of the mesh the velocity profiles converge to a finite curve shape around the distribution predicted for the 2450 element mesh.

In addition, the results were compared against results given by Reddy & Gartling [RG94]. This comparison is shown in Figure 4.35. It can be seen that the results are in very good agreement with Reddy & Gartling's data, especially at lower Reynolds-numbers. Their data was generated using a non-uniform, 14×14 mesh of linear rectangular elements. At higher

Reynolds-numbers the predictions at the lower half of the domain differ somewhat: however the trend follows. At the upper half of the domain closer to the sliding top lid, the results are more consistent with Reddy & Gartling's predictions. It should be noted that in the present analysis a uniform mesh was used with quadratic triangular elements for velocity. This results in additional velocity nodes between element vertices. The additional velocity nodes and using quadratic elements instead of linear elements results in increased accuracy, which would account for the advantages against the non-uniform mesh in Reddy & Gartling's calculations. The effect of the mesh size on predictions will be further discussed in Section 4.1.2.

For 50 elements the mesh is very coarse. The primary vortex is captured, which demonstrates the strength of the FEM method. The primary vortex is offset from the centre – as compared with Stokes vortices – due to the lower viscosity. The vortex moved slightly to the right and upward in the domain. The results for a 128 element domain are significantly more refined. While the characteristics of the flow pattern remained unchanged, one of the secondary vortices is captured by the solution (in the bottom right corner). Figures 4.36 and 4.37 show the streamtrace plots for discretizations with 392 and 2450 elements, respecitvely. With the increase of the mesh size, the definition of this primary vortex improved. For the last two mesh sizes (for 1800 and 2450 elements) the second secondary vortex in the lower left corner was also captured. As it will be seen later, these patterns are consistent with predictions by Teixeira [Tei97].

It was concluded that a discretization with 2450 elements would provide reasonably refined results, while keeping the computational time and resources low.

Temporal convergence

During temporal convergence testing the time step (Δt) is varied, while the discretization and the flow conditions are kept constant. The results are compared against each other. Temporal convergence is achieved when further reduction in the time step does not produce significant

Figure 4.33: Spatial convergence for lid-driven cavity flow (1)

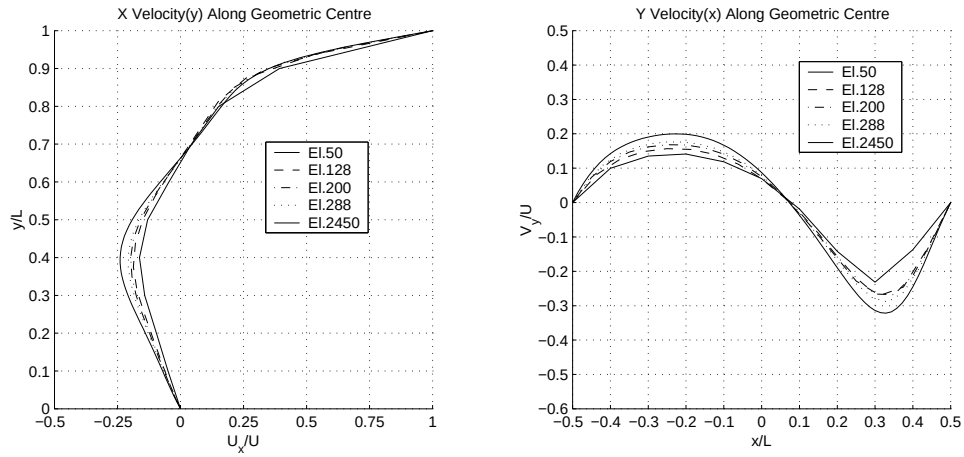


Figure 4.34: Spatial convergence for lid-driven cavity flow (2)

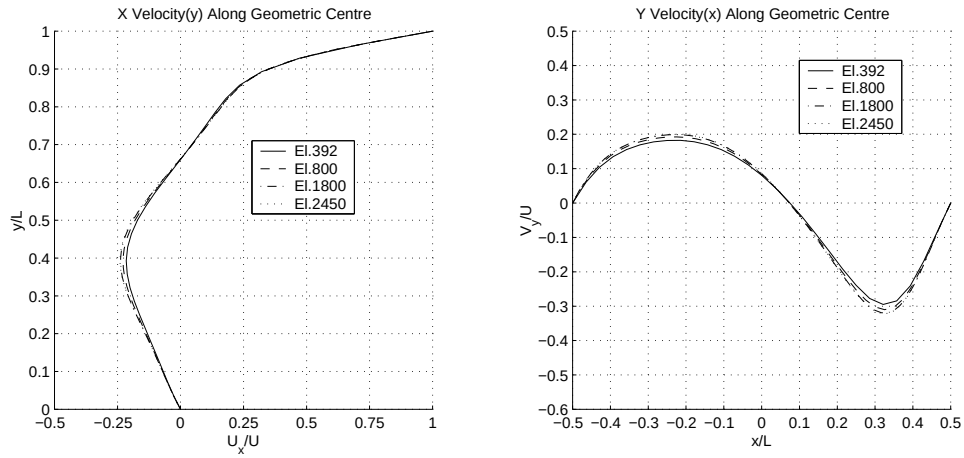
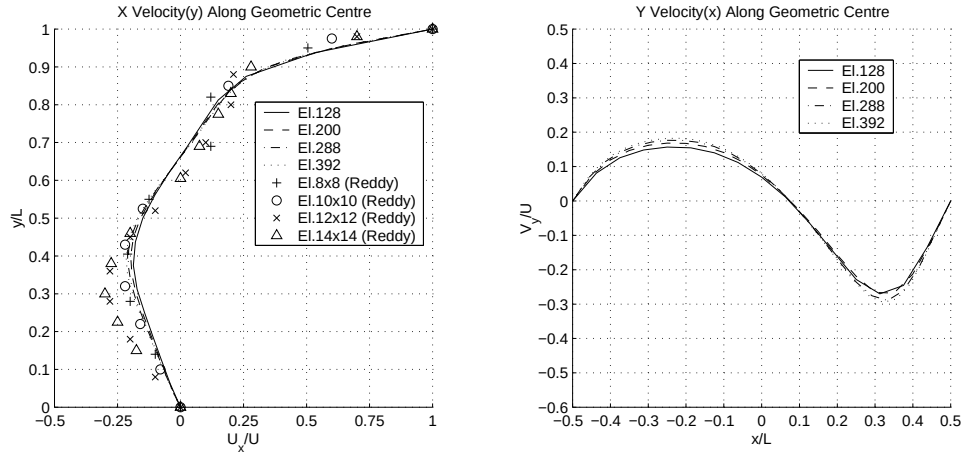


Figure 4.35: Spatial convergence for lid-driven cavity flow, comparison with [RG94]



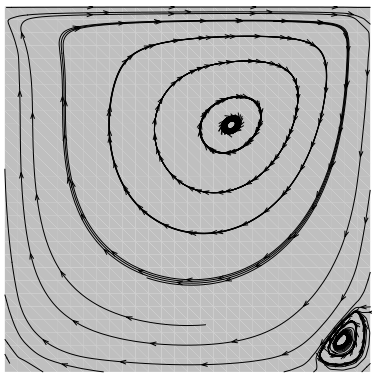


Figure 4.36: Navier-Stokes Streamtraces (392 elements)

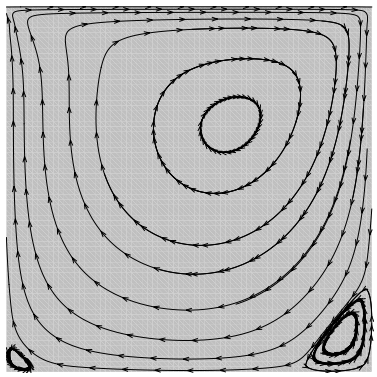


Figure 4.37: Navier-Stokes Streamtraces (2450 elements)

Table 4.14: Temporal convergence for a square cavity

Δt	*Iterations*	$t_{time\ series}$	*CPU time*
0.005	3628	18.14	2759.3
0.015	1745	26.175	1643.4
0.028	1099	30.772	1470.6
0.05	701	35.05	1153.5
0.1	*120*	*12*	*550.8*
0.5	*31*	*15.5*	*776*
1	*31*	*31*	*573.8*

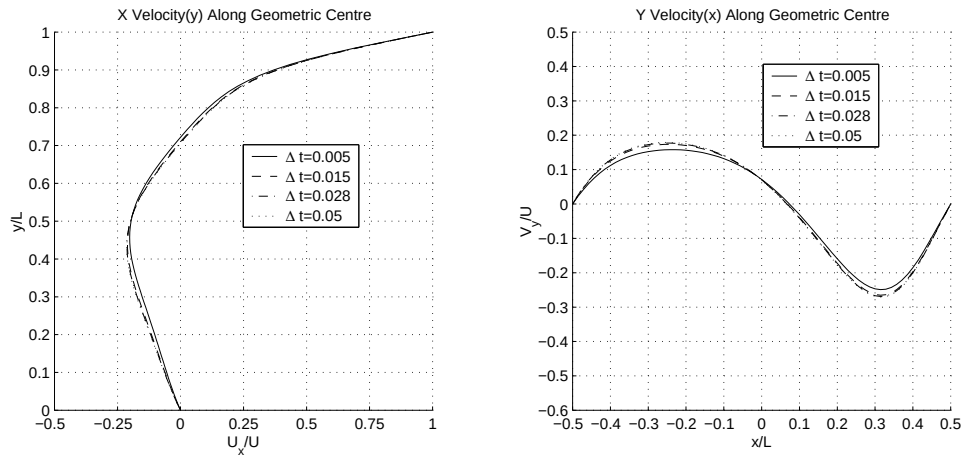
increase in the precision of the calculated variables.

The numerical testing of the Navier-Stokes solver for a square cavity is performed at a Reynolds-number of 400. The domain was broken down to 2450 elements. The convergence criterion was set to 5×10^{-5}. The time steps used in the iterations is given in Table 4.14.

The results for the temporal convergence tests are shown in Figures 4.38 and 4.39. It can be seen that the code is stable for time steps below the stability criteria and for a number of values above it, up to $\Delta t = 0.05$. It is about twice as high as the one calculated with Equation 3.58. This "discrepancy" can be easily explained. The critical time step was calculated from the smallest gap and the highest velocity. The highest velocity was conservatively equated with the velocity of the sliding top lid. However, as was shown by Moffatt [Mof64] and through the previous numerical test for triangular lid-driven Stokes flow (see Section 4.1.1, the flow velocity inside the domain is significantly lower than the sliding lid velocity, which would results in a higher Δt_{crit} than used here.

At higher time steps, such as for $\Delta t = 0.1$, 0.5, and 1, convergence was not reached.

Figure 4.38: Temporal convergence for lid-driven cavity flow



As expected, smaller time steps result in a faster convergence rates to reach the convergence criteria. In the stable region the RMS changes follow very similar trends for all predictions. It was found that there are no significant changes in the solutions at the same time series time using different time steps in the stable region. Runs performed at the lower time step range are considered more accurate, since at lower time steps the introduced accumulated numerical errors are smaller when an explicit numerical scheme is used, such as in this case.

Reynolds number dependence

The Navier-Stokes code was tested through a wide array of conditions to check the stability of the code and to test the validity of the predictions. Besides the spatial and temporal convergence tests the code was tested for a range of Reynolds-numbers for two different mesh sizes, such as for 2450 elements and for 392 elements. The first one was chosen as the finest discretization used during these calculations. The second was selected to allow for comparison with data by Reddy & Gartling [RG94]. The test conditions for 2450 elements are given in Table 4.15, while for 392 in Table 4.16. The time step was set according to the critical time step criteria (Eq.3.58), with an increased safety factor. Therefore, the actual time steps are shown in subsequent tables describing the run results.

Figure 4.39: Temporal convergence for lid-driven cavity flow - RMS changes

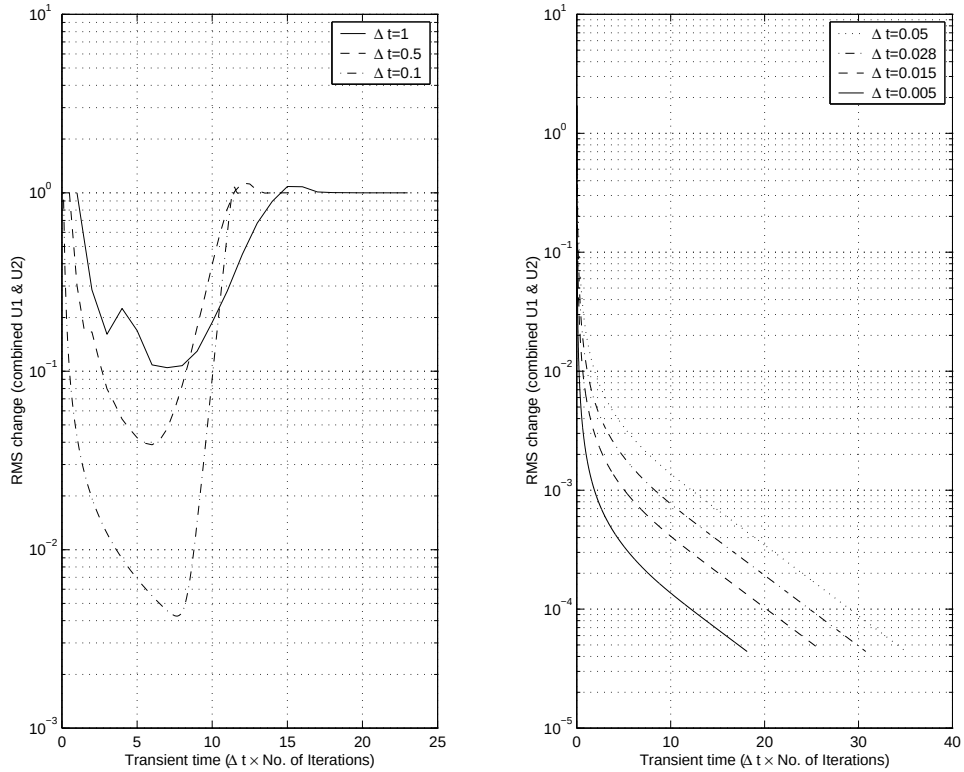


Table 4.15: Reynolds dependent test cases for 2450 elements

N_x	*Elem.*	U_{lid}	*Re*	Δt_{crit}	Δt_{used}	*Conv.Crit.*	*Iterations*	*CPUtime*
35	2450	0.33	1	0.028	0.005	$5E-05$	35	1190
35	2450	0.33	10	0.028	0.005	$5E-05$	220	690
35	2450	0.33	100	0.028	0.005	$5E-05$	1303	1158
35	2450	0.33	400	0.028	0.005	$5E-05$	3628	2759
35	2450	0.33	1000	0.028	0.005	$5E-05$	6490	5733
35	2450	0.33	2000	0.028	0.005	$5E-05$	8954	10461
35	2450	0.33	3200	0.028	0.005	$5E-05$	10763	13581

Table 4.16: Reynolds dependent test cases for 392 elements

N_x	*Elem.*	U_{lid}	*Re*	Δt_{crit}	Δt_{used}	*Conv.Crit.*	*Iterations*	*CPUtime*
14	392	0.5	1	0.046	0.02	$5E-05$	14	13.17
14	392	0.5	100	0.046	0.02	$5E-05$	466	37.18
14	392	0.5	400	0.046	0.02	$5E-05$	1427	140.77
14	392	0.5	1000	0.046	0.02	$5E-05$	2329	215.20

As an indication of the code's performance on the computer used for the numerical investigations, the CPU time for each run is also shown in Tables 4.15 and 4.16 for the 2450 and 392 elements, respectively. It can be seen from the number of iterations and the CPU times that for a significant increase in iteration numbers – using the same domain discretization – the CPU time does not increase proportionately. It was found that the largest computational effort is dedicated to the LU decomposition of the system matrix. The actual iteration steps required only back substitution, and therefore, minimal computational effort and CPU time.

2450 element domain

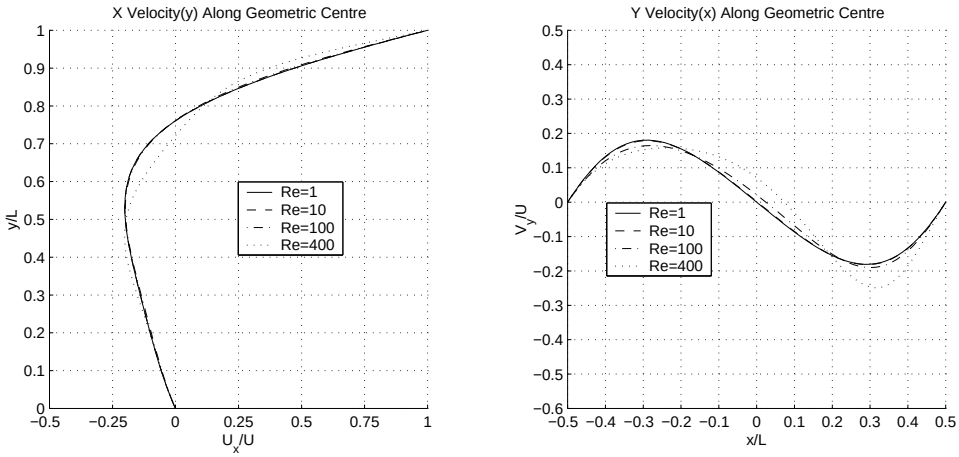
The results for a 2450 element domain are shown in Figures 4.40 and 4.41. The streamtrace plots are given in Figures 4.42 to 4.46.

For Re=1 (Figure 4.42) the solution captured the primary vortex in the center of the domain and 2 secondary vortices at the lower left and right corners of the cavity. As mentioned before, these are similar to Stokes vortices. Of course, looking at Equation 3.53, it can be seen that in the Navier-Stokes equations the time dependent term influences the system matrix through the Mass matrix. With the increase of the Reynolds-number the Stiffness matrix reduces and the Mass matrix becomes more and more dominant. Figures 4.42 to 4.46 show the effect of the increasing Reynolds-numbers on the flow pattern.

Up to Re=100 the velocity profiles and the velocity distribution do not change significantly. However, at Re=400 (Figure 4.43), the location of the primary vortex starts to shift which changes the velocity distribution symmetry. This shift to the direction of the sliding lid's moving direction along the positive x-axis also increases the size of the secondary vortex in the bottom right corner of the domain. With the increase of the Reynolds-number up to 3200 (see Figure 4.46) this secondary vortex increases significantly. Results by Teixeira [Tei97] indicate the development of a third secondary vortex at the upper left corner. (See also Appendix P for streamline patterns in a cavity at high Re numbers [She90].) While the present solution for a 2450 element mesh size did not capture this third secondary vortex, the flow pattern suggests its existence. The reason while this third vortex was not captured will be examined in the next subsection (Section 4.1.2).

The results from this analysis were compared against results by Teixeira [Tei97]. (Numerical results by Teixeira for lid-driven cavity flow is shown in Appendix P.) It was found that at low Reynolds-numbers the velocity profiles are in very good agreement. With the increase

Figure 4.40: Reynolds number dependency for 2450 elements for lid-driven cavity flow (1)

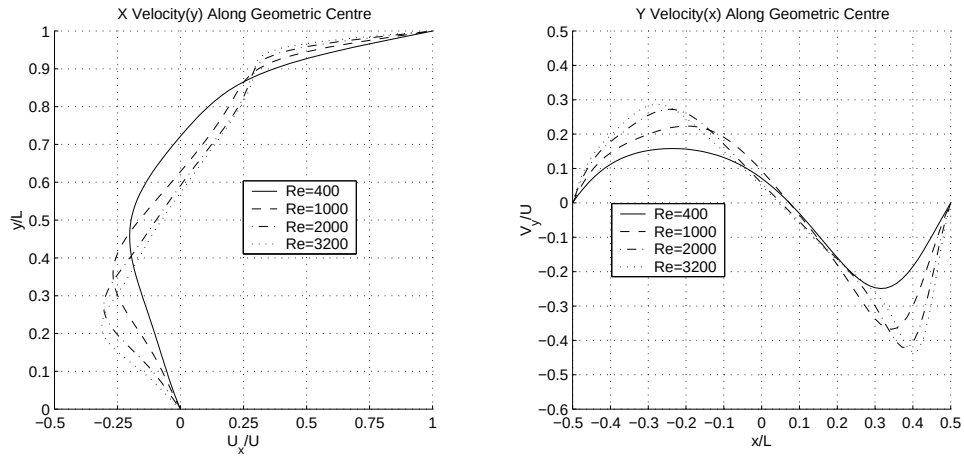


of the Reynolds-number the velocity distribution at the top half of the domain agrees well. However, at the lower half of the domain the present analysis predicts lower velocities close to the bottom wall of the cavity. The differences are attributed to the discretization. Teixeira used a discretization with 256×256 elements, while the current simulation was performed using $35 \times 35 \times 2$ elements (35×35 element blocks, with 2 triangular elements in each rectangular block). The effect of the mesh size and Reynolds numbers on the simulation results will be examined in Section 4.1.2. It should be noted that the trend of the velocity profile change was captured correctly.

392 element domain

Numerical tests were performed for a 392 element domain, to compare predictions with results by Reddy & Gartling [RG94]. The velocity profiles – including the data by Reddy & Gartling – are shown in Figure 4.47. It was found that the present results were in excellent agreement with previous results.

Figure 4.41: Reynolds number dependency for 2450 elements for lid-driven cavity flow (2)



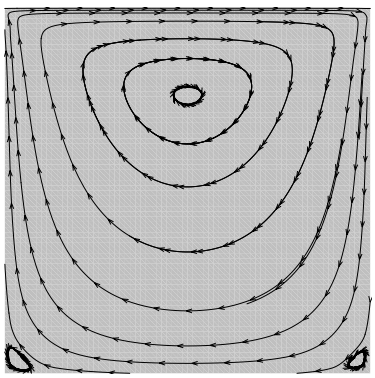


Figure 4.42: Streamtraces at Re=1 for 2450 elements

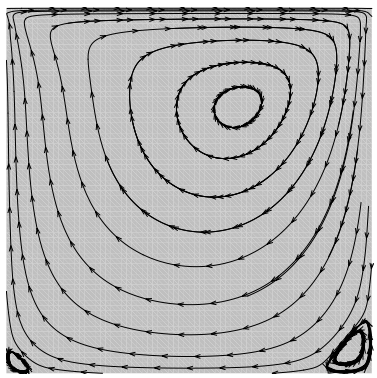


Figure 4.43: Streamtraces at Re=400 for 2450 elements

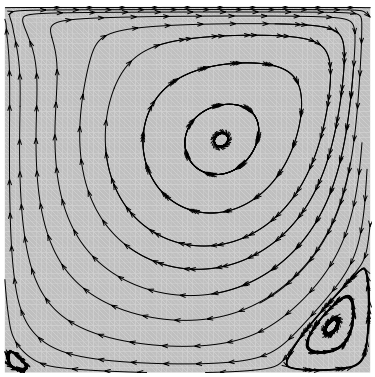


Figure 4.44: Streamtraces at Re=1000 for 2450 elements

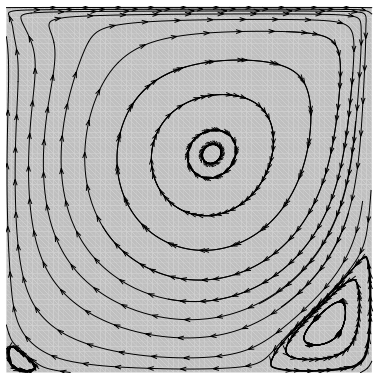


Figure 4.45: Streamtraces at Re=2000 for 2450 elements

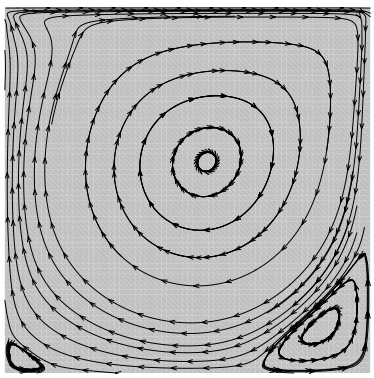
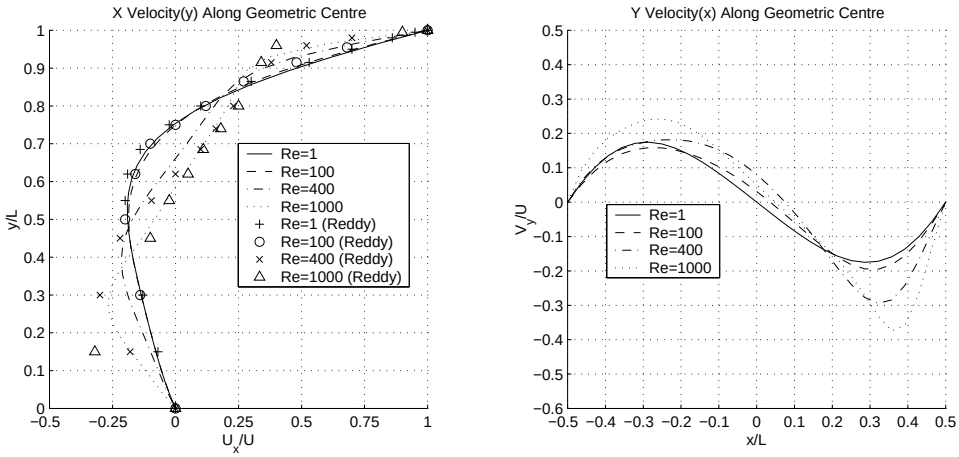


Figure 4.46: Streamtraces at Re=3200 for 2450 elements

Figure 4.47: Comparison with the results of Reddy & Gartling [RG94] for 392 elements



Considerations on discretization and boundary layers

The numerical example shown here for lid-driven cavity flow in a square domain is often used as a benchmark problem to test a numerical method or formulation. In solving this problem, the mesh used should be such that the boundary layer thickness is resolved. The boundary layer thickness is of the order of $Re^{-1/2}$, where $Re = \rho UL/\mu$ and U is the lid velocity and L is the cavity dimension. To achieve this the mesh can be refined or can be biased towards the wall.

This subsection examines the boundary-layer thickness for various mesh sizes and Reynolds-numbers.

Table 4.17 gives a summary of mesh sizes for various discretizations. For a square domain the number of elements blocks along the x- and y-axes are identical ($N_x = N_y$). When triangular elements are used (with two triangular element in a rectangular element block) the number of elements and the element size (h_x) can be calculated from N_x and N_y. When quadratic elements are used, there are nodes between the vertices of the triangular elements ($h' = h_x/2$).

Table 4.17: Element size for various discretizations

$N_x = N_y$	*Elements*	h_x	$h' = h_x/2$
1	2	1.00000	0.50000
2	8	0.50000	0.25000
3	18	0.33333	0.16667
4	32	0.25000	0.12500
5	50	0.20000	0.10000
8	128	0.12500	0.06250
10	200	0.10000	0.05000
12	288	0.08333	0.04167
14	392	0.07143	0.03571
15	450	0.06667	0.03333
20	800	0.05000	0.02500
25	1250	0.04000	0.02000
30	1800	0.03333	0.01667
35	2450	0.02857	0.01429
256	*131072*	*0.00391*	*0.00195*

Table 4.18: Boundary layer thickness relative to cavity size

For any number of elements						
$Re = 1$	$Re = 10$	$Re = 100$	$Re = 400$	$Re = 1000$	$Re = 2000$	$Re = 3200$
1.0000	0.3162	0.1000	0.0500	0.0316	0.0224	0.0177

The boundary layer thickness for various Reynolds-numbers is given in Table 4.18. From this and the mesh size it is possible to calculate the number of nodes which are within the boundary layer. That is, during the numerical simulations the number of nodes capturing the boundary layer are given in Table 4.19.

It can be seen that for Re=1, the 2450 element mesh would capture 70 points in the boundary layer, while for Re=3200 it would only calculate the conditions in the boundary layer for one nodal point. At lower discretizations the nodal points within the boundary layer would

Table 4.19: Number of values calculated inside the boundary layer

Elements	$Re = 1$	$Re = 10$	$Re = 100$	$Re = 400$	$Re = 1000$	$Re = 2000$	$Re = 3200$
2	2	*0.63*	*0.20*	*0.10*	*0.06*	*0.04*	*0.04*
8	4	1.26	*0.40*	*0.20*	*0.13*	*0.09*	*0.07*
18	6	1.90	*0.60*	*0.30*	*0.19*	*0.13*	*0.11*
32	8	2.53	*0.80*	*0.40*	*0.25*	*0.18*	*0.14*
50	10	3.16	1.00	*0.50*	*0.32*	*0.22*	*0.18*
128	16	5.06	1.60	*0.80*	*0.51*	*0.36*	*0.28*
200	20	6.32	2.00	1.00	*0.63*	*0.45*	*0.35*
288	24	7.59	2.40	1.20	*0.76*	*0.54*	*0.42*
392	28	8.85	2.80	1.40	*0.89*	*0.63*	*0.49*
450	30	9.49	3.00	1.50	*0.95*	*0.67*	*0.53*
800	40	12.65	4.00	2.00	1.26	*0.89*	*0.71*
1250	50	15.81	5.00	2.50	1.58	1.12	*0.88*
1800	60	18.97	6.00	3.00	1.90	1.34	1.06
2450	70	22.14	7.00	3.50	2.21	1.57	1.24
131072	512	161.91	51.20	25.60	16.19	11.45	9.05

decrease, resulting in an erosion of the precision of the calculations (see italicized values). For a comparison the mesh used by Teixeira [Tei97] is shown at the last line of the table. For a 256×256 element domain (assuming quadratic elements) the solution would capture 9 nodal points within the boundary layer. This is better than the results for 2450 elements at Re=100.

Therefore, for the duct flow calculations, for the present discretization, it is recommended to use as low Reynolds-numbers as possible, and in the future a non-uniform "biased" mesh towards the duct walls to better account for the boundary layer and consequently to increase simulation accuracy.

4.1.3 Conclusions

This section reported the unit testing of the Stokes and Navier-Stokes codes for lid-driven cavity flows, for triangular and rectangular domains.

During the testing both the spatial and temporal convergence of the calculations were examined, and a relatively large test matrix was used to determine the code behaviour and its stability. The stability was based on the Adams-Bashford 3 scheme. Critical time steps were calculated for various discretizations at constant initial and boundary conditions. The stability calculations were also reported here.

It was shown in the analysis for Stokes and Navier-Stokes lid-driven cavity flows that the primary vortex has velocities several orders of magnitude larger than those for subsequent vortices. This was in agreement with the findings of Moffatt [Mof64] for a triangular domain and with Reddy & Gartling [RG94] and Teixeira [Tei97] for a square domain. It was found that the code, developed for this analysis, performed according to the theory, and according to previous experimental measurements. (A few examples from open literature are given in Appendix P.)

In conclusion, the FEM CFD code was successfully tested for both Stokes and Navier-Stokes lid-driven flows for triangular and square domains.

4.2 Duct flows

4.2.1 Flow conditions for the test cases

The numerical test matrix for duct flow cases were performed for flow conditions relevant to inspiration. It is assumed that the flow is quasi-parallel and consequently the pressure is uniform across the flow. Typically, the maximal inspiratory flow rate is about 1 l/s [AD95]. Assuming a $D = 0.02m$ diameter, the cross section is about about 3 cm^2. The equivalent diameter of a two-dimensional duct is $D = 2h$ [Won77], from which the duct height is calculated to be $h = 0.01m$. At $300K$ the air density is $\rho_a = 1.1774\ kg/m^3$. The dynamic viscosity $\mu_a = 1.98 \times 10^{-5}\ kg/ms$ and the kinematic viscosity is $\nu_a = 1.68 \times 10^{-5}\ m^2/s$. ($\nu = \mu/\rho$) This

corresponds to a velocity of $v = 3.18\ m/s$, and a Reynolds-number of $Re = 3780$. Numerical simulations are performed at velocities and Reynolds numbers up to these maximum values. The pressure drop induced by friction losses along the soft palate is negligible in comparison to the inertial effects as long as the height of the channel is more that 1 mm.

4.2.2 Backward facing step

The laminar flow over a two-dimensional, backward facing step is frequently used as a standard test problem for numerical methods. A typical configuration is shown in Figure 4.48. Gartling used such a configuration to test outflow boundary conditions [Gar90]. Benchmark results were also documented by Reddy & Gartling [RG94]. The left boundary is an inflow and the right boundary is outflow. The bottom boundary is a solid wall, while the top boundary may be a wall or a boundary at which the flow field is essentially inviscid. For this particular case the top wall is chosen to be solid. This problem is also too easy a test, mainly because certain benign features of the flow, e.g., the position of the reattachment point behind the step, scales with the Reynolds-number [HH84].

The duct length was set to $L = 0.04m$, with a duct height of $h = 0.01m$. The streamwise length and duct height were divided into 72 and 20 element blocks, respectively. An 18 element long and 10 element high block was removed from the mesh to achieve the backward facing step geometry. This discretization resulted in a 2520 element domain. The test conditions are given in Table 4.20. A typical example of the results is shown in Figure 4.49, for Re=189 and Re=3780. The code correctly reproduced the expected trend of the flow circulation. The magnitude of the re-circulation zone is shorter than those in relevant literature. Freitas [Fre95] reported benchmark results from several commercial codes. Table 4.21 shows a comparison between benchmark results from Flow 3D and the present analysis. The difference between the results is attributed to the coarser discretization, and the fact that the present Navier–Stokes formulation is for laminar flows only, where the prediction accuracy is reduced with the increase of the Reynolds-number. For example, a Reynolds-number of 3780 would be in the

Figure 4.48: Backward facing step

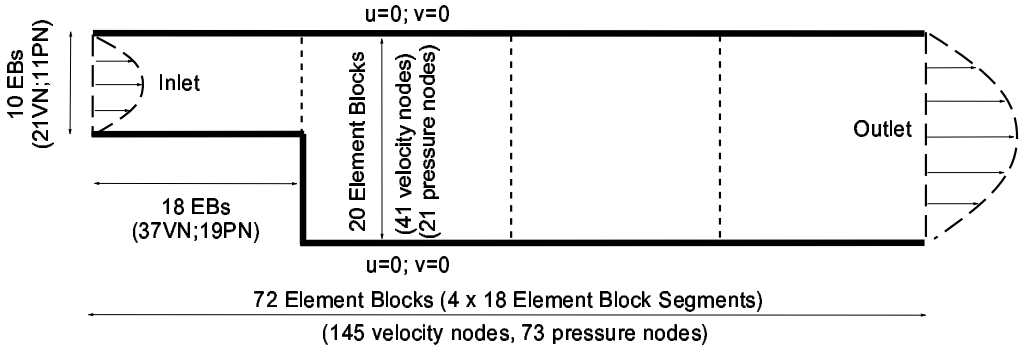


Table 4.20: Test matrix for the backward facing step cases

Case#	*Flow rate*	*Velocity*	*Re*	dt_{crit}	dt_{used}	RMS_{crit}	*Iter#*	*CPU*
$ST1$	$1.25E-5$	1	1	n/a	n/a	n/a	n/a	941
$ST2$	$1.25E-5$	0.1592	1	n/a	n/a	n/a	n/a	1026
$NS1$	$1.25E-5$	0.1592	94	0.00057	0.00028	$1.00E-05$	12	648
$NS2$	$2.5E-5$	0.3183	189	0.00028	0.00014	$1.00E-05$	29	610
$NS3$	$5.0E-5$	0.6366	378	0.00014	$7.09804E-05$	$1.00E-05$	90	483
$NS4$	$1.0E-4$	1.2732	756	0.00007	$3.54902E-05$	$1.00E-05$	329	550
$NS5$	$1.5E-4$	1.9099	1134	0.00005	$2.36601E-05$	$1.00E-05$	732	753
$NS6$	$2.0E-4$	2.5465	1512	0.00004	$1.77451E-05$	$1.00E-05$	1223	1140
$NS7$	$2.5E-4$	3.1831	1890	0.00003	$1.77451E-05$	$1.00E-05$	1337	1270

turbulent region, where the present formulation can not be used with confidence to reproduce accurate results. However, it is expected that the trend would match the expected flow trend. At low Reynolds-numbers the flow closely follows the duct geometries. Similar to Stokes flow conditions, the flow pattern does not exhibit re-circulation regions. At high Reynolds-numbers re-circulation zones appear, and with the increase of the flow velocity the zone length increases. This is consistent with theory.

Additional verification runs using basic unit testing geometries, such cases for Poiseuille type

Figure 4.49: Sample result for backward facing step

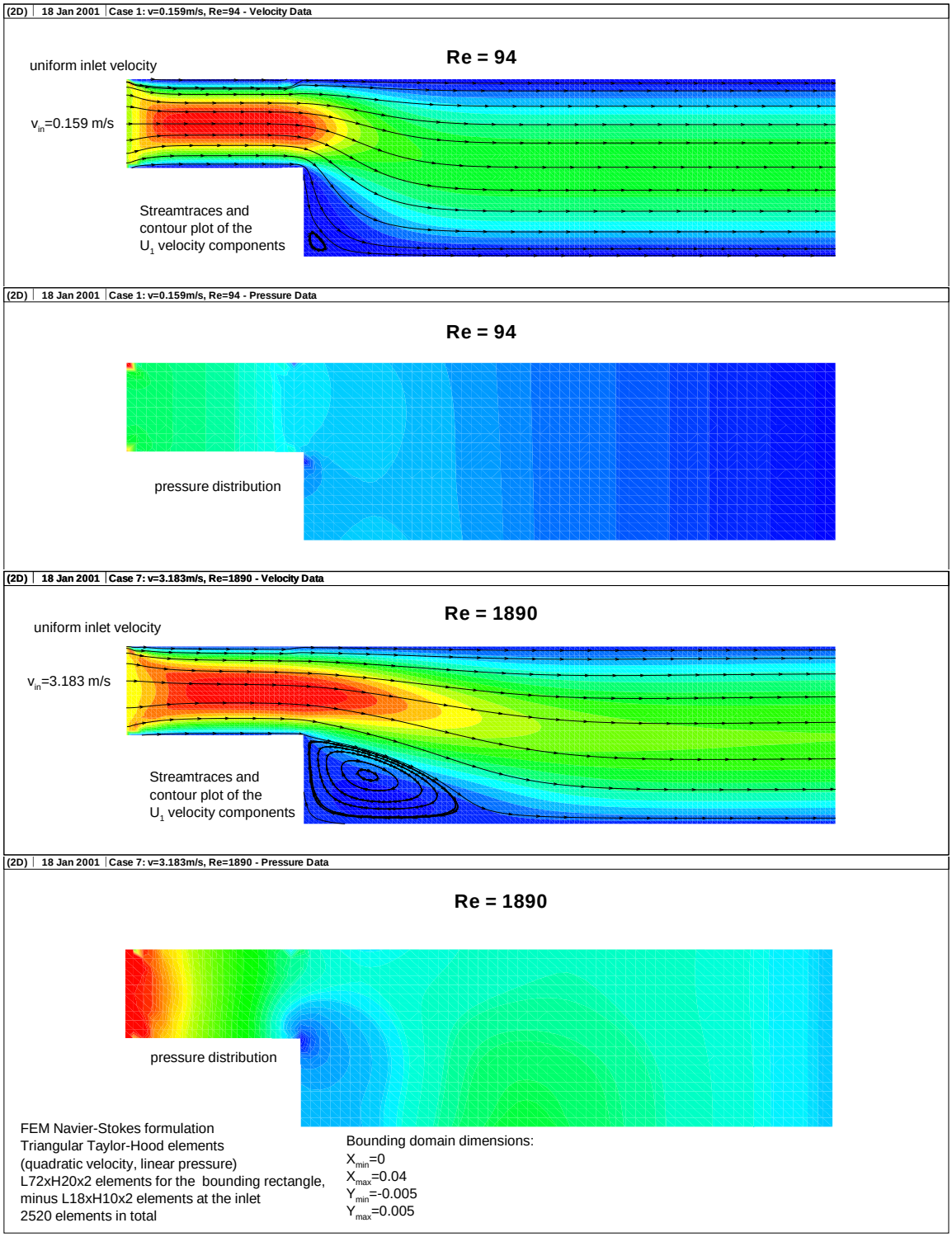


Table 4.21: Backward facing step, benchmark comparison with Freitas [Fre95]

	Re	200	450	1000
	v_{in}	0.15	0.33	0.74
$Re - attachment$	$x1$	0.0016	0.002	0.0024
Step height	s	0.00478	0.00478	0.004784211
Results	$x1/s$	0.334	0.418	0.502
Flow 3D/Freitas		*4.9*	*8.78*	*12.65*
Ratio		*14.65*	*21.00*	*25.22*

duct flow, for forward facing step and for fence in a duct are shown in Appendix N.

4.2.3 Duct with a split inlet

A simplified problem of the upper airway with rigid walls is an intermediate step towards modeling the upper airway with a flexible wall segments. Figure 4.50 shows the configuration for this model. With the two-inlet configuration the fluid enters through two channels at the left side of the domain with uniform velocity, and exits to the right. Due to the no-slip boundary condition a parabolic velocity profile develops by the time the flow reaches the exit. The $L = 0.04m$ domain is divided into 80 element blocks, while the height ($h = 0.01m$) is divided into 16 element blocks, resulting in a 2560 element domain. The wall, which splits the inlet, is 30 element blocks long, leaving 50 elements for the combined regions. For the second configuration we consider the lower inlet blocked.

Two-inlet configuration

The test conditions are given in Table 4.22. A typical example of the results are shown in Figure 4.51, for Re=189 and Re=3780. Following the uniform inlet of the flow the parabolic velocity profile develops rapidly. At the end of the split the two branches join together forming a single channel flow. At low Reynolds-numbers this redistribution region is very short, while at high Reynolds-numbers it is stretched out. The pressure distribution is similar to the expected

Figure 4.50: Rectangular channel with split inlet

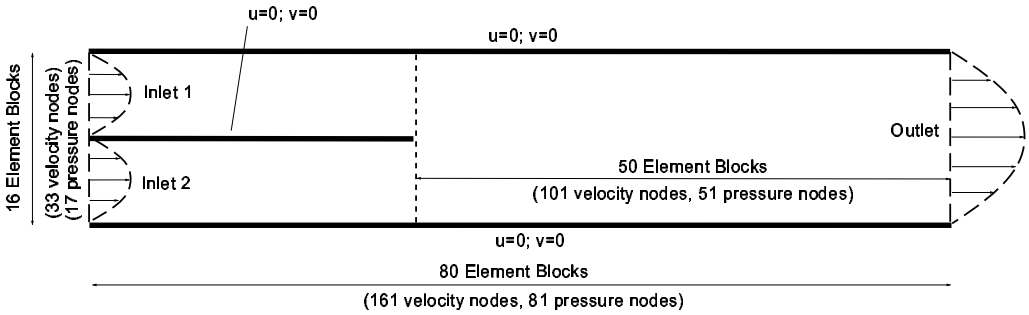


Table 4.22: Test matrix for the horizontal split with two inlets cases

$Case\#$	$Flow\ rate$	$Velocity_{in}$	Re_{in}	dt_{crit}	dt_{used}	RMS_{crit}	$Iter\#$	CPU
$ST1$	0.00005	1	1	n/a	n/a	n/a	n/a	412
$ST2$	0.00005	0.1592	1	n/a	n/a	n/a	n/a	411
$NS1$	0.00005	0.1592	189	$71.0E-05$	$35.5E-05$	0.00001	12	447
$NS2$	0.0001	0.3183	378	$35.5E-05$	$17.7E-05$	0.00001	31	394
$NS3$	0.0002	0.6366	756	$17.7E-05$	$8.9E-05$	0.00001	91	433
$NS4$	0.0004	1.2732	1512	$8.9E-05$	$4.4E-05$	0.00001	295	426
$NS5$	0.0006	1.9099	2268	$5.9E-05$	$3.0E-05$	0.00001	461	568
$NS6$	0.0008	2.5465	3024	$4.4E-05$	$2.2E-05$	0.00001	539	619
$NS7$	0.001	3.1831	3780	$3.5E-05$	$1.8E-05$	0.00001	736	787

duct flow pressure distributions.

One-inlet configuration – blocked lower inlet

When the lower inlet is blocked, the flow enters through the upper inlet. The test conditions are given in Table 4.23. Typical examples of the results are shown in Figure 4.52, for Re=189 and Re=3780. This configuration can be viewed as a combination of a backward facing step and a lid driven cavity flow case. The flow enters from the left side of the domain with uniform velocity and develops a parabolic cross sectional velocity profile. When it reaches the end of the plate dividing the channel, the flow expands and a new parabolic velocity profile develops. However, shear layer at the edge of the flow between the tip of the horizontal plate and the

Figure 4.51: Sample result for a horizontal split with two inlets

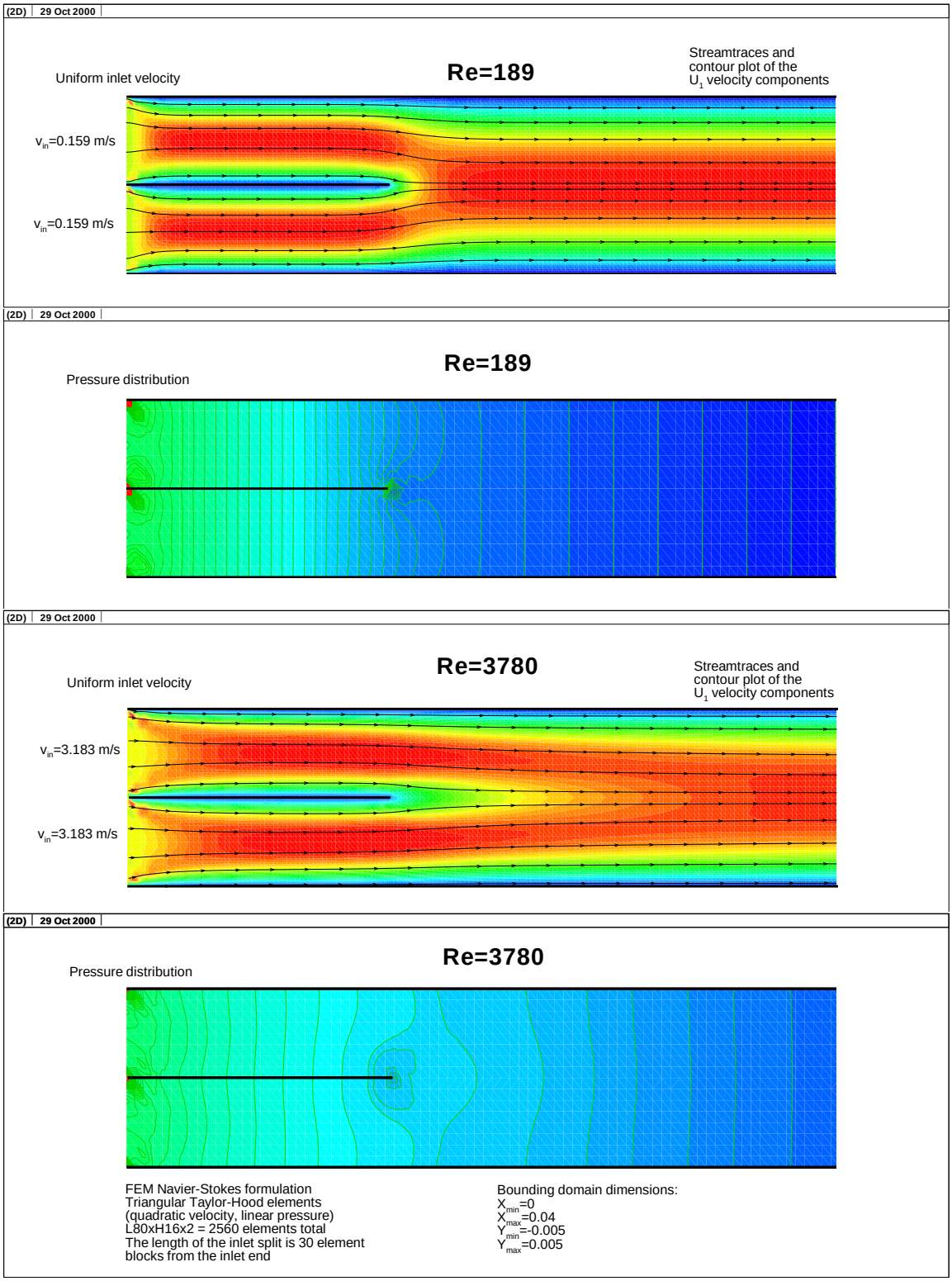


Table 4.23: Test matrix for the horizontal split with one inlet cases

$Case\#$	$Flow\ rate$	$Velocity_{in}$	Re_{in}	dt_{crit}	dt_{used}	RMS_{crit}	$Iter\#$	CPU
$ST1$	$1.25E-5$	1	1	n/a	n/a	n/a	n/a	690
$ST2$	$1.25E-5$	0.1592	1	n/a	n/a	n/a	n/a	690
$NS1$	$1.25E-5$	0.1592	94	$71.0E-05$	$35.5E-05$	0.00001	11	636
$NS2$	$2.5E-5$	0.3183	189	$35.5E-05$	$17.7E-05$	0.00001	26	489
$NS3$	$5.0E-5$	0.6366	378	$17.7E-05$	$8.9E-05$	0.00001	77	428
$NS4$	$1.0E-4$	1.2732	756	$8.9E-05$	$4.4E-05$	0.00001	278	549
$NS5$	$1.5E-4$	1.9099	1134	$5.9E-05$	$3.0E-05$	0.00001	608	654
$NS6$	$2.0E-4$	2.5465	1512	$4.4E-05$	$2.2E-05$	0.00001	1000	939
$NS7$	$2.5E-4$	3.183	1890	$3.5E-05$	$1.8E-05$	0.00001	1349	1150

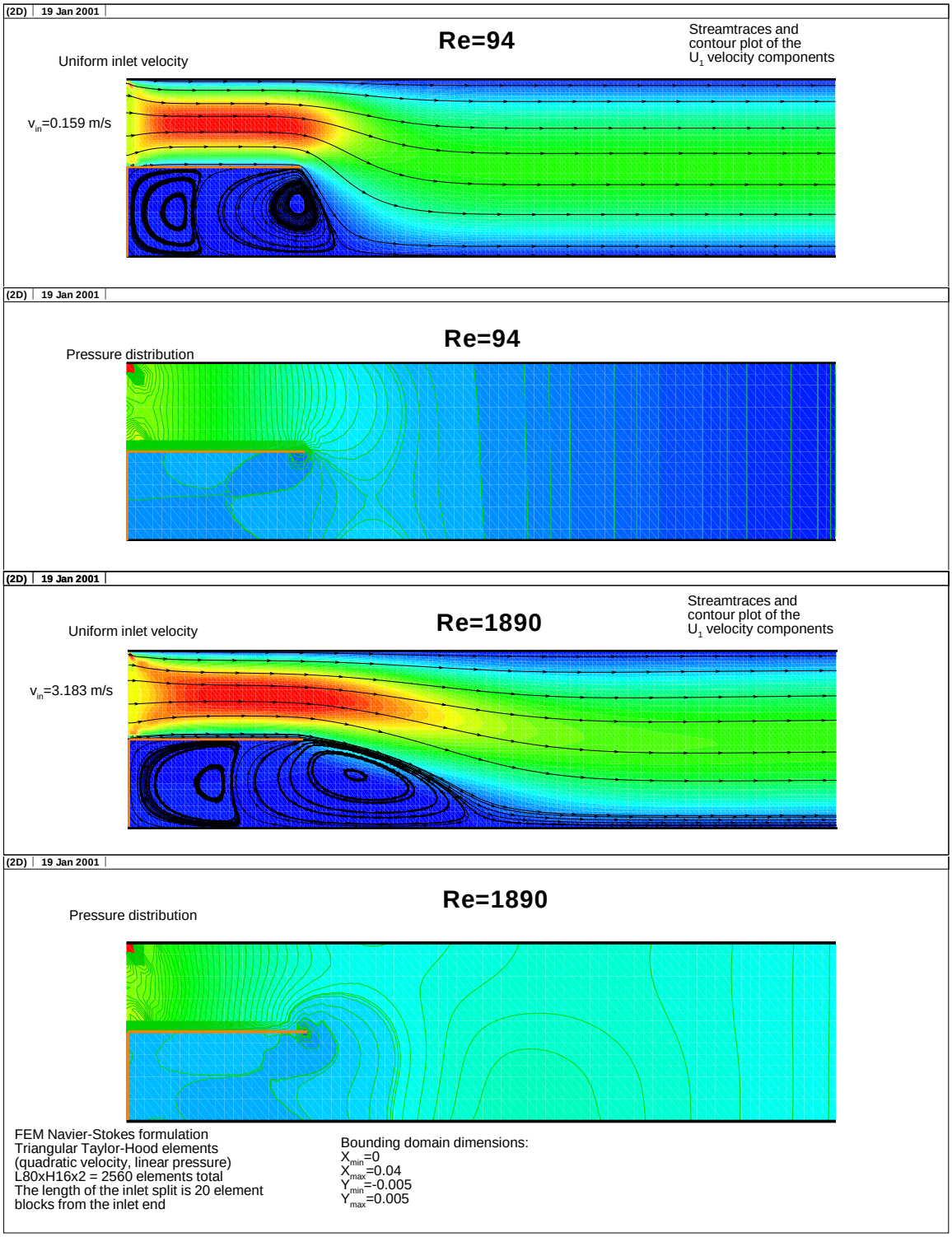
duct wall act like the sliding lid shown earlier. The solution captures two counter-rotating vortices in the cavity between the closed inlet wall and the end tip of the separation plate. At low Reynolds-numbers the shear layer is almost perpendicular to the wall. As the flow velocity increases, the shear layer becomes more inclined resulting in an elongated first vortex.

4.2.4 Upper airway representation

4.2.5 Model of the upper airway

Finite element model of supraglottal vocal tract was given by [DHV+99], in which the upper airway geometries between the mouth and the larynx were reported, as shown in Figure 4.53. The data is shown in Table 4.24. The considered geometry and volume of this acoustic space corresponds to the English vowel "a". The hydraulic diameter for a pipe is the same as its true diameter. Therefore, the diameters reported in the table are the same as the hydraulic diameters for the "pipe". The present work models the vocal tract as a two dimensional duct. The hydraulic diameter of a two-dimensional duct is twice the duct height ($D_h = 2h$), from which the duct height can be calculated at each axial location. The 2D model is shown in Figure 4.54. Another assumption was made regarding the oral and nasal cavities. During normal breathing the mouth is closed to a point, where the diameters for the oral and nasal

Figure 4.52: Sample result for a horizontal split with one inlet



openings are approximately the same. To achieve the same flow rate through the mouth and nose, the local resistance must be the same. Therefore, it is assumed that the same geometry of the upper airway can be used for the split upper airway as the one given in [DHV+99], by artificially dividing the duct into an upper and a lower half. Note that the lower figure shows the divided airway using a solid section followed by a dashed line. The solid line represents the hard palate, while the dashed line indicates the soft palate. Following the initial inlet, the oral and nasal cavities open up then around the soft palate narrow back. The narrowest section corresponds to the pharyngeal region. There is also a difference at the exit between the 3D and 2D geometries. The data of [DHV+99] indicates an expansion before the exit plane is reached. The 2D duct model uses a straight exit. This does not change the results significantly, but it avoids further numerical complications.

The $L = 0.17458m$ domain is divided into 80 element blocks, while the height is divided into 16 element blocks, resulting in a 2560 element domain. The wall, which splits the inlet, is 37 element blocs long, leaving 43 elements for the combined regions. (This corresponds to a 2.76 node length/node height ratio.) For the second configuration we consider the lower inlet blocked.

Two inlets

The test conditions are given in Table 4.25. Typical examples of the results are shown in Figure 4.55, for Re=189 and Re=3780. Note that the critical time step was calculate based on the inlet velocity. However, due to the narrowing of the duct, the local maximum velocity is significantly larger. Therefore, for the critical velocity $1/10th$ of the calculated value was used. The code correctly captured the trend for velocity and pressure distributions. The highest velocities occurred at the narrowest region, corresponding to the pharynx. It should be noted that for such a long domain this discretization is rather coarse. In future work it is recommended to use a numerical method which allows for higher nodalization.

Table 4.24: Upper airway and equivalent duct geometries

#	*Coord.*	$R(m)$	$D(m)$	$h(m)$	#	*Coord.*	$R(m)$	$D_h(m)$	$h(m)$
1	0	0.01265	0.0253	*0.01265*	23	0.08932	0.00578	0.01156	*0.00578*
2	0.00406	0.01222	0.02444	*0.01222*	24	0.09338	0.00618	0.01236	*0.00618*
3	0.00812	0.01166	0.02332	*0.01166*	25	0.09744	0.00458	0.00916	*0.00458*
4	0.01218	0.01163	0.02326	*0.01163*	26	0.1015	0.00357	0.00714	*0.00357*
5	0.01624	0.01147	0.02294	*0.01147*	27	0.10556	0.00298	0.00596	*0.00298*
6	0.0203	0.0111	0.0222	*0.0111*	28	0.10962	0.00304	0.00608	*0.00304*
7	0.02436	0.01223	0.02446	*0.01223*	29	0.11368	0.00319	0.00638	*0.00319*
8	0.02842	0.01296	0.02592	*0.01296*	30	0.11774	0.00271	0.00542	*0.00271*
9	0.03248	0.01375	0.0275	*0.01375*	31	0.1218	0.00299	0.00598	*0.00299*
10	0.03654	0.01413	0.02826	*0.01413*	32	0.12586	0.00288	0.00576	*0.00288*
11	0.0406	0.01415	0.0283	*0.01415*	33	0.12992	0.00352	0.00704	*0.00352*
12	0.04466	0.01444	0.02888	*0.01444*	34	0.13398	0.00448	0.00896	*0.00448*
13	0.04872	0.01384	0.02768	*0.01384*	35	0.13804	0.0052	0.0104	*0.0052*
14	0.05278	0.01273	0.02546	*0.01273*	36	0.1421	0.00597	0.01194	*0.00597*
15	0.05684	0.0121	0.0242	*0.0121*	37	0.14616	0.00578	0.01156	*0.00578*
16	0.0609	0.01093	0.02186	*0.01093*	38	0.15022	0.00324	0.00648	*0.00324*
17	0.06496	0.0098	0.0196	*0.0098*	39	0.15428	0.00309	0.00618	*0.00309*
18	0.06902	0.00954	0.01908	*0.00954*	40	0.15834	0.00319	0.00638	*0.00319*
19	0.07308	0.00941	0.01882	*0.00941*	41	0.1624	0.00259	0.00518	*0.00259*
20	0.07714	0.00903	0.01806	*0.00903*	42	0.16646	0.00288	0.00576	*0.00288*
21	0.0812	0.00816	0.01632	*0.00816*	43	0.17052	0.00252	0.00504	*0.00252*
22	0.08526	0.00718	0.01436	*0.00718*	44	0.17458	0.00378	0.00756	**0.00252**

Table 4.25: Test matrix for the vocal tract with two inlet cases

Case#	*Flow rate*	$Velocity_{in}$	Re_{in}	dt_{crit}	dt_{used}	RMS_{crit}	*Iter.#*	*CPU*
$ST1$	0.00005	1	1	n/a	n/a	n/a	n/a	505
$ST2$	0.00005	0.1592	1	n/a	n/a	n/a	n/a	433
$NS1$	0.00005	0.1592	189	$89.8E-05$	$9.0E-05$	0.00001	19	670
$NS2$	0.0001	0.3183	378	$44.9E-05$	$4.5E-05$	0.00001	53	658
$NS3$	0.0002	0.6366	756	$22.4E-05$	$2.2E-05$	0.00001	152	638
$NS4$	0.0004	1.2732	1512	$11.2E-05$	$1.1E-05$	0.00001	464	706
$NS5$	0.0006	1.9099	2268	$7.5E-05$	$7.5E-06$	0.00001	916	950
$NS6$	0.0008	2.5465	3024	$5.6E-05$	$5.6E-06$	0.00001	1435	1287
$NS7$	0.001	3.1831	3780	$4.5E-05$	$3.0E-06$	0.00001	2581	2268

Figure 4.53: Vocal tract geometries [DHV+99]

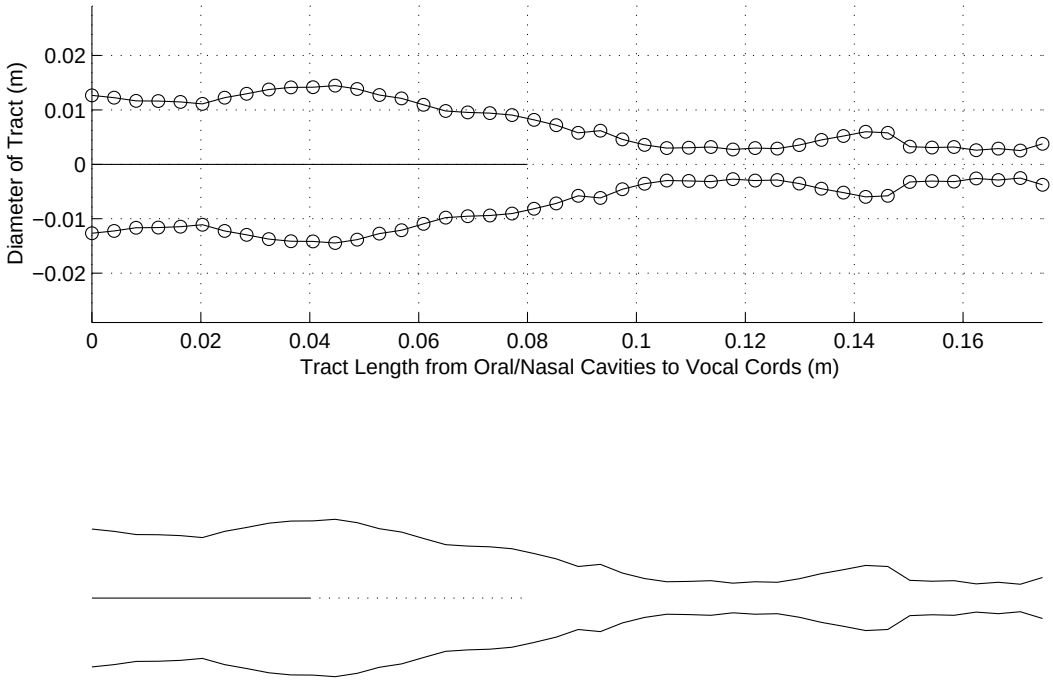
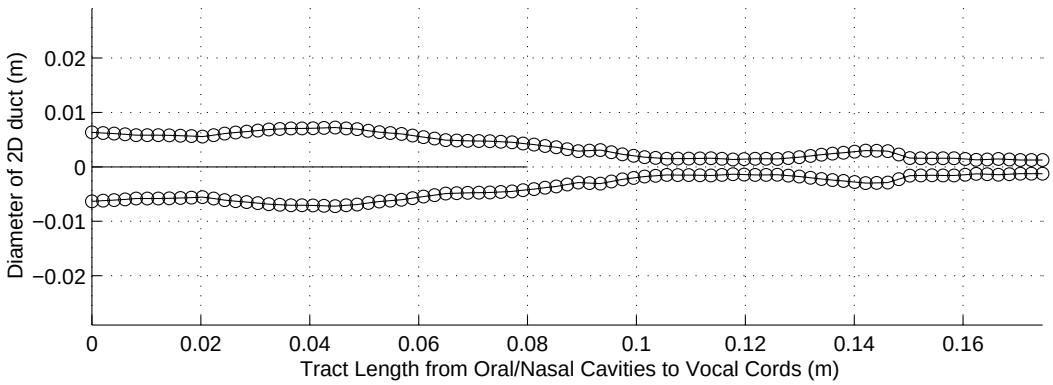


Figure 4.54: 2D duct model of the vocal tract



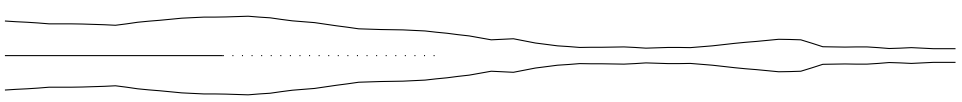


Figure 4.55: Sample result for a rigid vocal tract with two inlets

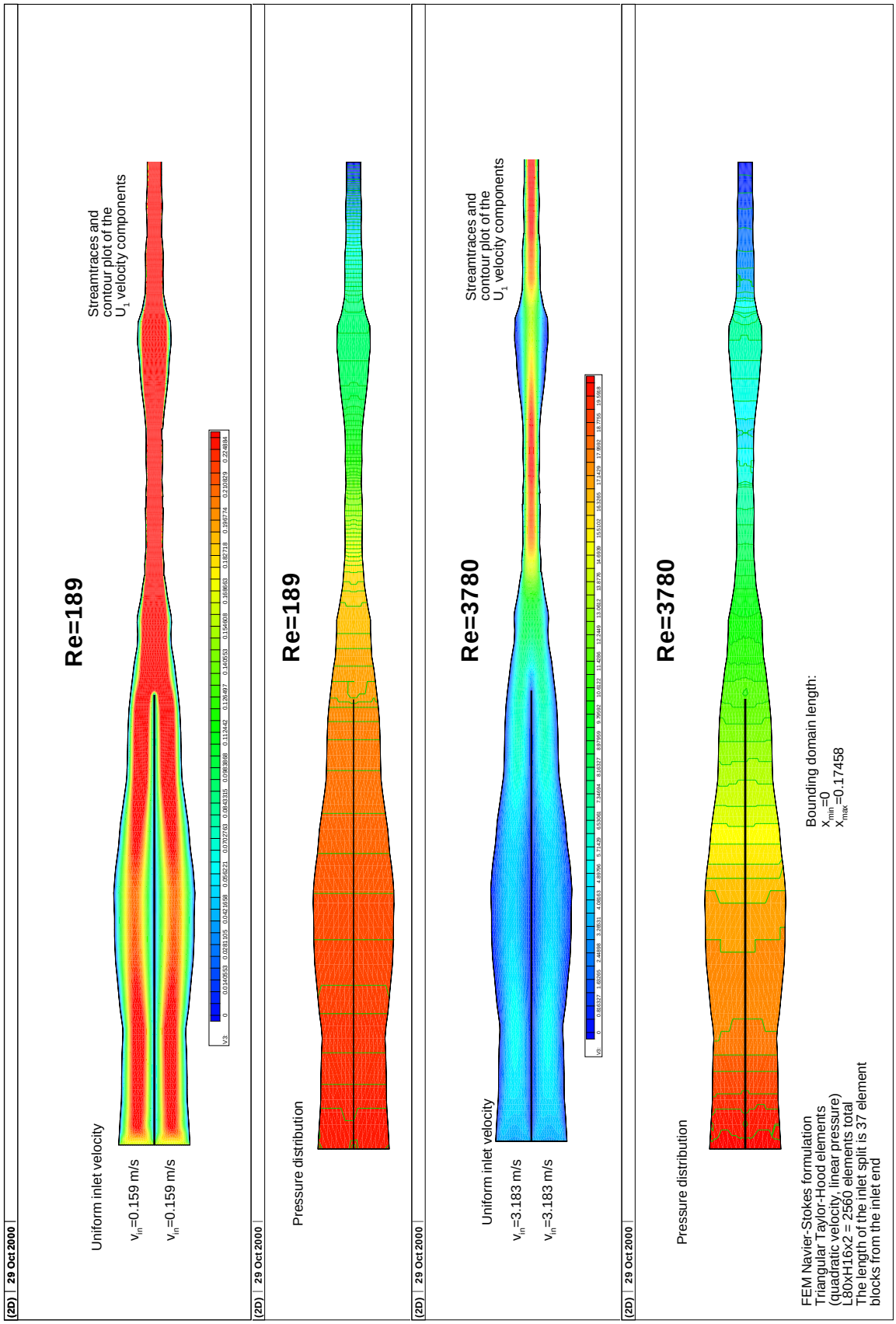


Table 4.26: Test matrix for the vocal tract with one inlet cases

$Case\#$	$Flow\ rate$	$Velocity_{in}$	Re_{in}	dt_{crit}	dt_{used}	RMS_{crit}	$Iter.\#$	CPU
$ST1$	$1.25E-5$	1	1	n/a	n/a	n/a	n/a	575
$ST2$	$1.25E-5$	0.1592	1	n/a	n/a	n/a	n/a	433
$NS1$	$1.25E-5$	0.1592	94	$89.8E-05$	$9.0E-05$	0.00001	16	660
$NS2$	$2.5E-5$	0.3183	189	$44.9E-05$	$4.5E-05$	0.00001	42	706
$NS3$	$5.0E-5$	0.6366	378	$22.4E-05$	$2.2E-05$	0.00001	136	626
$NS4$	$1.0E-4$	1.2732	756	$11.2E-05$	$1.1E-05$	0.00001	460	908
$NS5$	$1.5E-4$	1.9099	1134	$7.5E-05$	$7.5E-06$	0.00001	850	1050
$NS6$	$2.0E-4$	2.5465	1512	$5.6E-05$	$5.6E-06$	0.00001	1277	1390
$NS7$	$2.5E-4$	3.1831	1890	$4.5E-05$	$3.0E-06$	0.00001	2343	2170

One-inlet configuration – blocked lower inlet

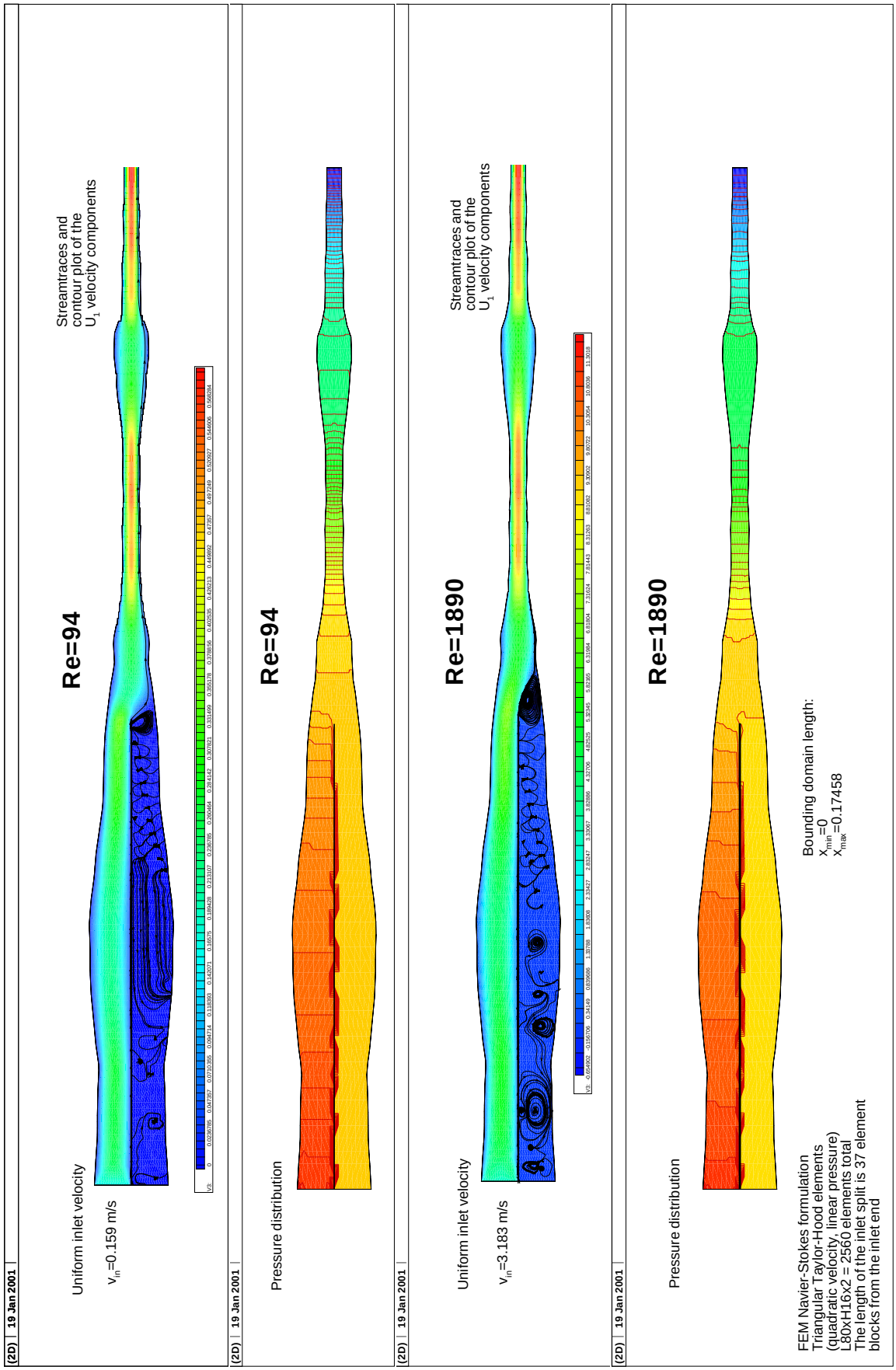
When the lower inlet is blocked, the situation is similar to the split inlet case explained in Section 4.2.3. The test conditions are given in Table 4.26. Typical examples of the results are shown in Figure 4.56, for Re=189 and Re=3780. The flow enters from the upper inlet, accelerates slightly then as the nasal cavity opens up, it decelerates. The highest velocity regions are at the pharyngeal and exit regions, where the diameter is the narrowest. However, the discretization is not fine enough to resolve the vortices in the closed off cavity.

4.2.6 Conclusions

Unit and benchmark cases were performed for duct flow cases using various configurations. These were: a backward facing step, a horizontally split duct with one or two inlets and a simplified model of the upper airway with one or two inlets. The results for Poiseuille type duct flow, forward facing step and fence in a duct are shown in Appendix N.

It was found that the code correctly predicted the flow features, such as the expected velocity and pressure profiles. When compared with other benchmark results, the difference in the magnitude of reattachment lengths was attributed to coarse discretization. Also, the present formulation of the Navier–Stokes equations is for laminar flow conditions. At higher Reynolds-

Figure 4.56: Sample result for a rigid vocal tract with one inlet



numbers this formulation becomes inappropriate.

In summary it was concluded that the code captured the expected flow behaviour and it is suitable for investigating the fluid-structure interaction using a coupled code, where the present Navier–Stokes solver is coupled with the the code predicting the displacement of a finite flexible plate.

Chapter 5

Validation of the flexible plate code

This chapter documents a computer code validation exercise for the flexible plate code, where the code was validated against analytical (eigenmode) solutions. This step was necessary to establish credibility for the flexible plate code before code coupling.

5.1 Analytical eigenmode solutions

5.1.1 Initial and boundary conditions

In vacuo the initial and boundary conditions are specified by the displacement of the plate between the two end points. The plate is discretized into a given number of nodes (from 1 to N) and at each nodal location a corresponding local plate displacement is assigned.

The motion of the plate in vacuo, without damping can is determined by

$$\rho_m h \frac{\partial^2 w}{\partial t^2} + B \frac{\partial^4 w}{\partial x^4} = 0 \tag{5.1}$$

5.1.2 Hinged-hinged initial and boundary conditions in vacuo

To test the code with hinged boundary conditions, pure displacement modes were generated by the equation

$$w = w_o e^{i\omega t} sin\left(\frac{n\pi x}{L}\right) \tag{5.2}$$

where w is the local displacement (m), w_o is the maximum displacement (m), ω is the frequency $(1/s)$, n is the mode number, x is the axial location along the plate (m) and L is the length of the plate (m).

From Equations 5.2 and 5.1 after differentiation

$$\underbrace{\left[-\rho_m h\omega^2 + B\left(\frac{n\pi}{L}\right)^4\right]}_{=0} w_o e^{i\omega t} sin\left(\frac{n\pi x}{L}\right) = 0 \tag{5.3}$$

from which the radian frequency of the disturbance is

$$\omega = \left[\frac{B\left(\frac{n\pi}{L}\right)^4}{\rho_m h}\right]^{1/2} = \frac{2\pi}{T} \tag{5.4}$$

The time period per cycle (in seconds) is given as

$$T = \frac{2\pi}{\omega} \tag{5.5}$$

The first six eigenvalues and eigenfunctions are shown in Figure 5.1 for a hinged–hinged plate.

5.1.3 Clamped-clamped initial and boundary conditions in vacuo

To test the code under clamped boundary conditions, eigenfunctions given by Geveci [Gev99] and Walker [WZD97] were used, after converting from their dimensionless form to dimensional form used throughout the present plate displacement calculations.

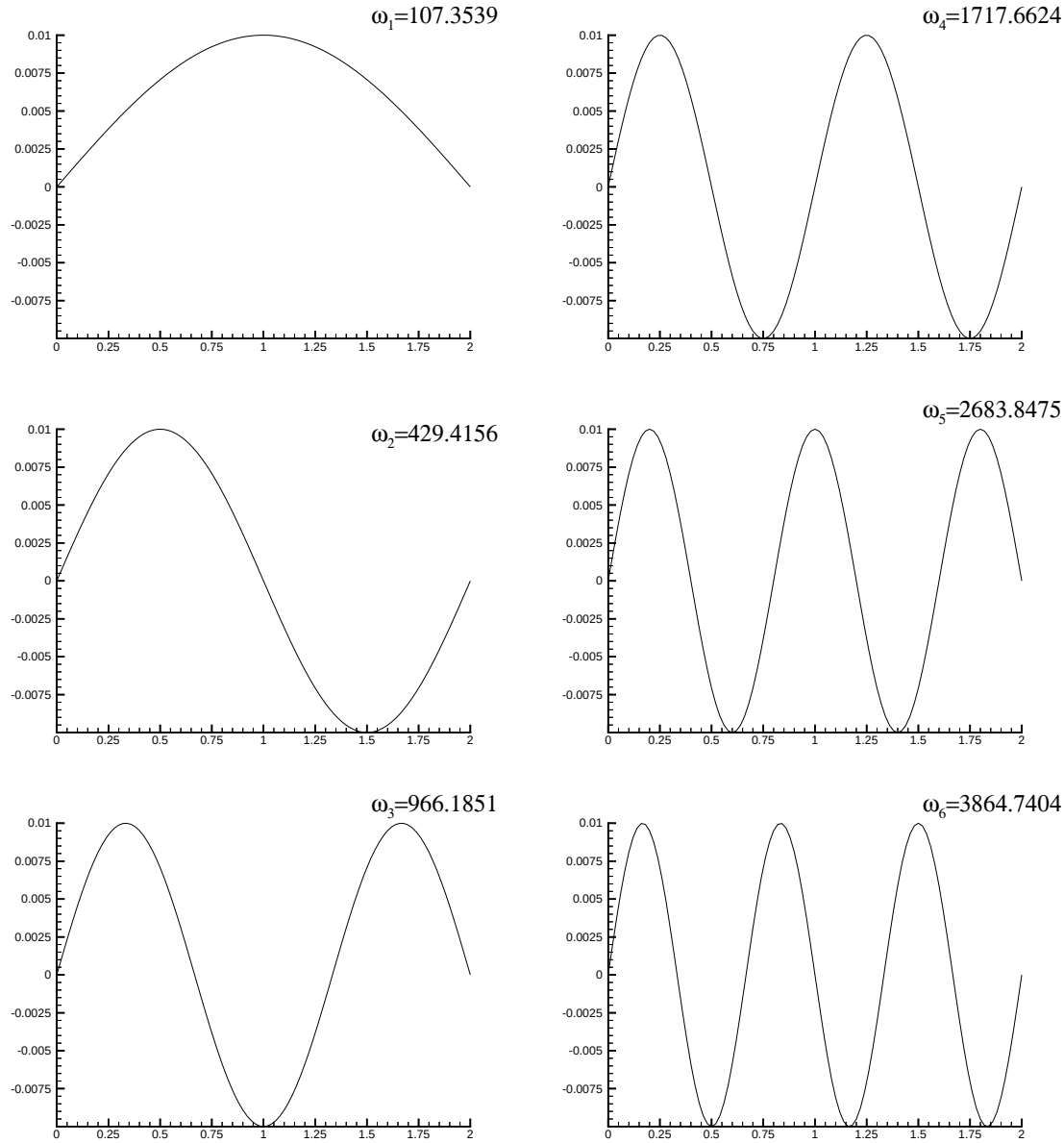
The relationship between the dimensional $(x, w, t \ldots)$ and dimensionless $(\bar{x}, \bar{w}, \bar{t} \ldots)$ variables are described below for axial location (x), displacement (w) and time (t)

$$x = L\,\bar{x} \tag{5.6}$$

$$w = h\,\bar{w} \tag{5.7}$$

$$t = \frac{L}{\epsilon}\sqrt{\frac{\rho_m h^3}{B}}\bar{t} \tag{5.8}$$

Figure 5.1: The first six eigenvalues and eigenfunctions for a hinged–hinged plate



where L is the plate length, h is the plate thickness, and $\epsilon = h/L$. Substituting Equations 5.6, 5.7 and 5.8 to Equation 5.1

$$\begin{aligned}
\rho_m h \frac{\partial^2(h\bar{w})}{\partial\left(\frac{L^2}{h}\sqrt{\frac{\rho_m h^3}{B}}\bar{t}\right)^2} + B\frac{\partial^4(h\bar{w})}{\partial(L\bar{x})^4} &= 0 \\
\frac{\rho_m h^2}{L^4\frac{\rho_m h}{B}}\frac{\partial^2\bar{w}}{\partial\bar{t}^2} + B\frac{h}{L^4}\frac{\partial^4(\bar{w})}{\partial(\bar{x})^4} &= 0 \\
B\frac{h}{L^4}\frac{\partial^2\bar{w}}{\partial\bar{t}^2} + B\frac{h}{L^4}\frac{\partial^4(\bar{w})}{\partial(\bar{x})^4} &= 0 \\
\underbrace{\frac{\partial^2\bar{w}}{\partial\bar{t}^2}}_{I} + \underbrace{\frac{\partial^4(\bar{w})}{\partial(\bar{x})^4}}_{II} &= 0
\end{aligned} \tag{5.9}$$

The solutions are sought in the form of

$$\bar{w}(\bar{x},\bar{t}) = \Re[X(\bar{x})e^{i\bar{\omega}\bar{t}}] \tag{5.10}$$

where the eigenvalues, for a beam clamped at both ends, satisfy

$$cos(\bar{\omega}^{1/2})\ cosh(\bar{\omega}^{1/2}) = 1 \tag{5.11}$$

Substituting Equation 5.10 to the first term (I) of Equation 5.9

$$\begin{aligned}
\frac{\partial^2\bar{w}}{\partial\bar{t}^2} &= \frac{\partial^2 X(\bar{x})e^{i\bar{\omega}\bar{t}}}{\partial\bar{t}^2} \\
&= X(\bar{x})\frac{\partial^2 e^{i\bar{\omega}\bar{t}}}{\partial\bar{t}^2} \\
&= X(\bar{x})i\bar{\omega}\frac{\partial e^{i\bar{\omega}\bar{t}}}{\partial\bar{t}} \\
&= -X(\bar{x})\bar{\omega}^2 e^{i\bar{\omega}\bar{t}}
\end{aligned} \tag{5.12}$$

Substituting Equation 5.10 to the second term (II) of Equation 5.9

$$\frac{\partial^4(\bar{w})}{\partial(\bar{x})^4} = e^{i\bar{\omega}\bar{t}}\frac{\partial^4 X(\bar{x})}{\partial(\bar{x})^4} \tag{5.13}$$

combining the resulting two parts back together

$$\begin{aligned}
-X(\bar{x})\bar{\omega}^2 e^{i\bar{\omega}\bar{t}} + e^{i\bar{\omega}\bar{t}}\frac{\partial^4 X(\bar{x})}{\partial(\bar{x})^4} &= 0 \\
\underbrace{\left(\frac{\partial^4 X(\bar{x})}{\partial(\bar{x})^4} - X(\bar{x})\bar{\omega}^2\right)}_{=0} e^{i\bar{\omega}\bar{t}} &= 0
\end{aligned} \tag{5.14}$$

This agrees with the dimensionless form of the equation of motion given by Geveci and Walker

$$\frac{d^4X(\bar{x})}{d\bar{x}^4} - \bar{\omega}^2 X(\bar{x}) = 0 \tag{5.15}$$

The corresponding eigenfunctions are given by Walker *et.al.* [WZD97] in the form of

$$X_i(\bar{x}) = J(\bar{\omega}^{1/2}\bar{x}) - \frac{J(\bar{\omega}^{1/2})}{H(\bar{\omega}^{1/2})} H(\bar{\omega}^{1/2}\bar{x}) \tag{5.16}$$

where

$$\begin{aligned} J(\,) &= cosh(\,) - cos(\,) \\ H(\,) &= sinh(\,) - sin(\,) \end{aligned} \tag{5.17}$$

The eigenvalues ($\bar{\omega}_n^{1/2}$) can be calculated using the expansion (Walker *et.al.* [WZD97]

$$\bar{\omega}_n^{1/2} = (2n+1)\frac{\pi}{2} + (-1)^{n+1}b_o - b_o^2 + \left\{\frac{1}{3}(-1)^{3n} + 2(-1)^{n+1}\right\} b_o^3 + \ldots \tag{5.18}$$

where $b_o = 1/cosh\{(2n+1)\pi/2\}$ and n is the mode number.

The dimensional time is given by Geveci and Walker as

$$T = \frac{L^2}{h}\sqrt{\frac{\rho_m h^3}{B}}\bar{T} = L^2\sqrt{\frac{\rho_m h}{B}}\bar{T} \tag{5.19}$$

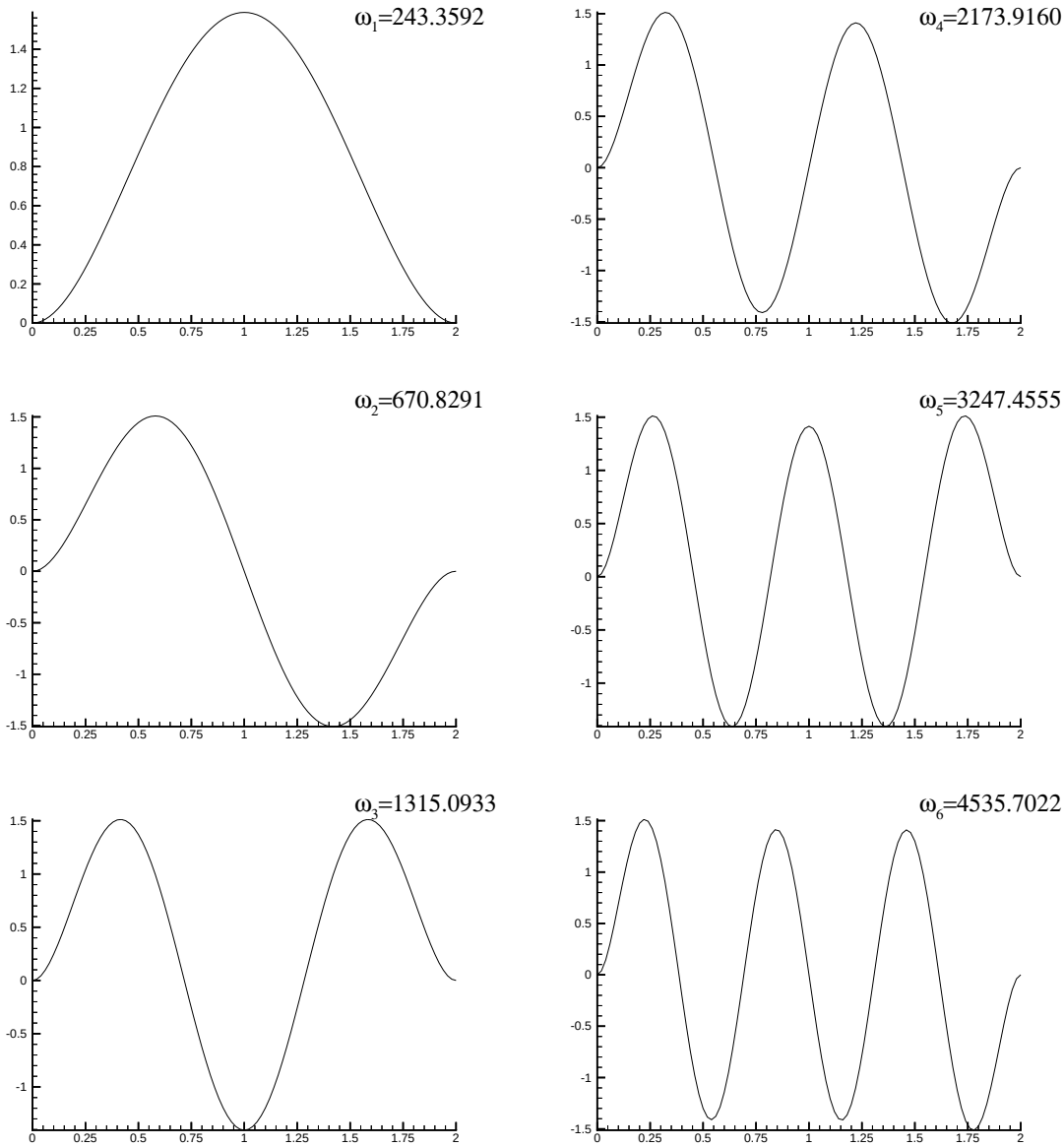
From Equations 5.19 and 5.5 the dimensional frequency is

$$\begin{aligned} \omega &= \frac{2\pi}{T} = \frac{2\pi}{L^2\sqrt{\frac{\rho_m h}{B}}\bar{T}} \\ \omega &= \frac{1}{L^2\sqrt{\frac{\rho_m h}{B}}}\frac{2\pi}{\bar{T}} \\ \omega &= \frac{\bar{\omega}}{L^2}\sqrt{\frac{B}{\rho_m h}} \end{aligned} \tag{5.20}$$

Once the dimensional frequency is calculated from Equations 5.18 and 5.20, the dimensional time period per cycle can be evaluated using Equation 5.5.

The first six eigenvalues and eigenfunctions are shown in Figure 5.2 for a clamped–clamped plate.

Figure 5.2: The first six eigenvalues and eigenfunctions for a clamped–clamped plate



5.1.4 Cantilevered initial and boundary conditions in vacuo

Transverse vibration of a rod is described by Nowacki [Now63]. For a rod clamped at $x = 0$ and free at $x = L$ the end conditions are

$$w(0) = \frac{\partial w(0)}{\partial x} = \frac{\partial^2 w(L)}{\partial x^2} = \frac{\partial^3 w(L)}{\partial x^3} = 0 \tag{5.21}$$

this results in a system of homogeneous equations. Details of the derivation are given by Nowacki. Equating to zero the principal determinant of the last system of equations (not shown here), the obtained transcendental equation is

$$cosh\beta\ cos\beta + 1 = 0 \tag{5.22}$$

the roots of which are

$$\begin{aligned} \beta_1 &= 1.875 \\ \beta_2 &= 4.694 \\ \beta_3 &= 7.855 \\ \beta_4 &= 10.996 \\ \beta_5 &= 14.137 \end{aligned} \tag{5.23}$$

and

$$\beta_n = \frac{2n-1}{2}\,\pi \qquad \text{for} \quad n > 5 \tag{5.24}$$

The form of the fundamental mode vibrations, from which the initial nodal displacement values are calculated, is the following:

$$w_n(x) = C\left[U(\lambda_n x) - \frac{S(\beta_n)}{T(\beta_n)} V(\lambda_n x)\right] \qquad n = 1, 2, 3 \ldots \infty \tag{5.25}$$

where C is a scaling constant, λ_n is

$$\lambda_n = \frac{\beta_n}{L} \tag{5.26}$$

and V, U, T and S are

$$V(\lambda_n x) = \frac{sinh(\lambda_n x) - sin(\lambda_n x)}{2} \tag{5.27}$$

$$U(\lambda_n x) = \frac{cosh(\lambda_n x) - cos(\lambda_n x)}{2} \tag{5.28}$$

$$T(\beta_n) = \frac{sinh(\beta_n) + sin(\beta_n)}{2} \tag{5.29}$$

$$S(\beta_n) = \frac{cosh(\beta_n) + cos(\beta_n)}{2} \tag{5.30}$$

The consecutive frequencies of vibration have the form

$$\omega_n = \frac{\beta_n^2}{L^2}\sqrt{\frac{B}{\rho_m h}} \qquad n = 1, 2, 3 \ldots \infty \tag{5.31}$$

The first six eigenvalues and eigenfunctions are shown in Figure 5.3.

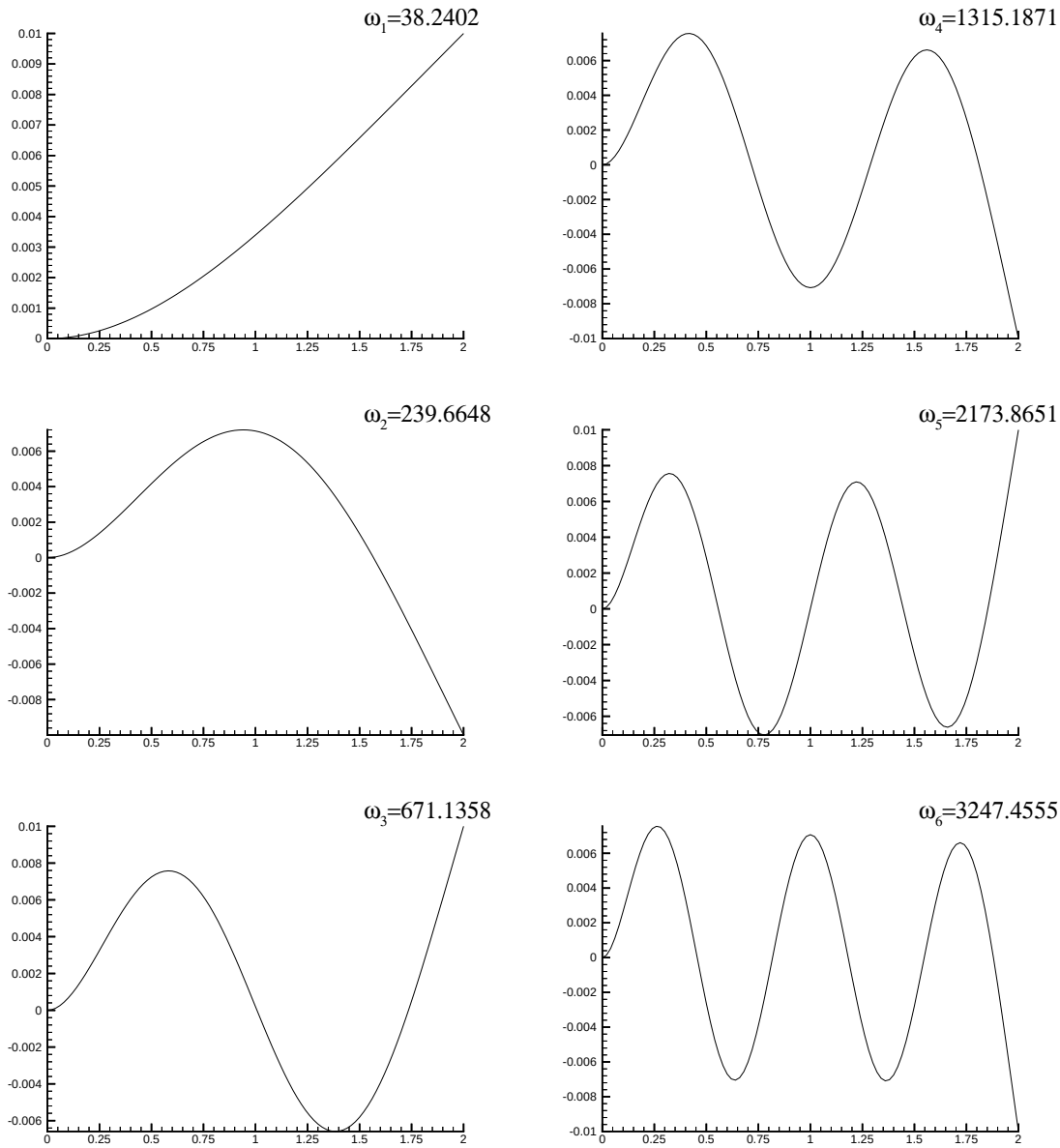
5.2 Numerical simulations

5.2.1 Methodology and assumptions

The main program was developed using FORTRAN 90/95. The initial nodal displacements were pre-calculated in MS–Excel spreadsheets or using an input generator written in FORTRAN. These pre-calculations allowed for the generation of fundamental mode displacements for any of the end conditions examined, based on basic input variables (e.g., mode number, density, Young's modulus, Poisson ratio, plate thickness and length). The variables used for the two–ends fixed conditions have some physical validity, representing the characteristics of common glass [Luc89]. However, it should be noted that these calculations were performed to test and validate the code against analytical solutions under various configurations and boundary conditions, and not to examine specific material behaviours.

While in the first part the Navier–Stokes equations for 2D duct flow were programmed using an explicit FEM method, the equations of motion for the 2D moving plate were programmed using an implicit finite difference method. Details of the equations and the boundary conditions are given in Section 3.4. For the elastic plate, a linear model was used. Fluid loading and coupling between the fluid and the wall codes will be considered in Chapter 6.

Figure 5.3: The first six eigenvalues and eigenfunctions for a cantilevered plate



Verification of the computer programs were performed by "stepping through" the codes, while monitoring key variables. The Debugger of the FORTRAN compiler was used for this verification step.

In the numerical simulations first the flexible plate was displaced using various initial eigenfunction based displacement shapes. From this plate displacement position the plate was released, then the nodal displacement, velocity and acceleration values were calculated through the time series.

The accuracy of the calculations was measured by how well the analytical and numerical solutions matched after a predefined number of time series steps. A numerical prediction was considered accurate, if the nodal displacements at the end of the time series matched the analytical solution.

The code was validated using fundamental modes in vacuo. However, the program can also account for uniform or non-uniform damping and pressure distribution along the plate.

Throughout these calculations the plate was discretized to 101–102 nodal points, plus 2–4 "dummy" points outside the domain, based on the end conditions of the specific configuration. The output from the main program was formatted in a way, which allowed for easy graphical post–processing with the TecPlot plotting package. The discretization of the domain does not account for the plate thickness. The length of the plate was set to non-unit length to test the code under more complex geometries.

The convergence was determined by satisfying an RMS condition. The Root Mean Square condition for the implicit scheme was considered satisfied, when the summed up RMS error – calculated between the old (guessed) and subsequently calculated new acceleration nodal

values – reached a given limit. The RMS condition was calculated from

$$RMS = \sqrt{\frac{\sum (\ddot{w}_{new} - \ddot{w}_{old})^2}{\sum \ddot{w}_{new}^2}} \leq RMS_W \tag{5.32}$$

The RMS convergence criterion was set to $RMS_W = 1 \times 10^{-12}$ for all cases in the test matrices.

5.2.2 Hinged–hinged plate fixed at both ends

The test matrices for the hinged–hinged configuration are given in Tables 5.1 and 5.2 for fundamental modes in vacuo, in Table 5.3 for in vacuo with damping and in Table 5.4 for conditions with pressure and damping. For all cases the elastic modulus, Poisson ratio, density, plate thickness and maximum displacement were set to $E = 5.25 \times 10^{+13}\ N/m^2$, $\nu = 0.33333$, $\rho = 2600\ kg/m^3$, $h = 0.001\ m$ and $w_0 = 0.01\ m$, respectively.

The first displacement cycle for all six eigenmodes are shown in Figure 5.4.

Figure 5.5 gives the oscillations for Mode 1 (a) in vacuo without damping, (b) in vacuo with uniform damping and (c) with uniform pressure and damping. The first mode peaks in the middle of the plate (Location 0.5) and due to symmetry, Locations 0.3 and 0.7 overlap in the time series runs (see figures (a) to (c)). The two ends of the plate are fixed, therefore the displacements at Locations 0.0 and 1.0 are zero. When damping is applied, the amplitude of the harmonic wave reduces towards zero as shown in figure (b). When pressure is accounted for, the converging displacement curve is offset from zero. Due to the uniform pressure force, the relative displacement order of the nodal points remain unchanged. That is, the mid-point has the largest displacement, the end points remains zero, and the intermediate points keep their relative order between the maximum and minimum locations.

Figure 5.6 gives the oscillations for Mode 2 (a) in vacuo without damping, (b) in vacuo with damping and (c) with pressure and damping. The second mode has a peak at Location 0.25 and a trough at Location 0.75. This shows up as a phase offset on the figures. Locations

Table 5.1: Test matrix for hinged–hinged plates in vacuo

	$Case1$ $Mode1$	$Case2$ $Mode2$	$Case3$ $Mode3$	$Case4$ $Mode4$	$Case5$ $Mode5$	$Case6$ $Mode6$
n	1	2	3	4	5	6
L	2	2	2	2	2	2
B	4921.86	4921.86	4921.86	4921.86	4921.86	4921.86
ω	107.35	429.41	966.18	1717.66	2683.85	3864.74
$T = 2\pi/\omega$	0.0585	0.0146	0.0065	0.0036	0.0023	0.0016
Δt	$1E-6$	$1E-6$	$1E-6$	$1E-6$	$1E-6$	$1E-6$
$\#\ of\ sets$	100	100	100	100	100	100
$print\_step$	585	146	65	36	23	16

0.0 and 1.0 are fixed. In vacuo the plate at Location 0.5 is also not moving, since for this mode shape the middle of the plate is stationary throughout the time series. However, when a constant pressure force is applied, Location 0.5 shows a secondary wave that settles with time due to damping and can be controlled with the damping coefficient. The final damped steady state curve in figure (c) resembles a Mode 1 displacement curve, where Location 0.5 represents the peak, Locations 0.0 and 1.0 are zero and the intermediate points are set before these maximum and minimum values.

Figure 5.7 gives the oscillations for Mode 3 (a) in vacuo without damping, (b) in vacuo with damping and (c) with pressure and damping. The mode three curve shows similar behaviour to the Mode 2 curve in the sense of the phase offsets. Location 0.5 points to the bottom of the trough with the largest magnitude of displacement. Locations 0.3 and 0.7 are coinciding and are offset relative to Location 0.5. The behaviour shown in figures (b) and (c) are similar to the one explained for Mode 2.

Table 5.2: Test matrix for hinged–hinged plates in vacuo (long runs)

	$Case1$ $Mode1$	$Case2$ $Mode2$	$Case3$ $Mode3$
n	1	2	3
L	2	2	2
B	4921.86	4921.86	4921.86
ω	107.35	429.41	966.18
$8 \times T$	0.4682	0.1170	0.0520
$\#\ of\ sets$	200	200	200
$print\_step$	2341	585	260

Table 5.3: Test matrix for hinged–hinged plates in vacuo with damping

	$Case1$ $Mode1$	$Case2$ $Mode2$	$Case3$ $Mode3$
n	1	2	3
L	2	2	2
B	4921.86	4921.86	4921.86
d	50	150	450
ω	107.35	429.41	966.18
$8 \times T$	0.4682	0.1170	0.0520
Δt	$1E-6$	$1E-6$	$1E-6$
$\#\ of\ sets$	200	200	200
$print\_step$	2341	585	260

Figure 5.4: The first six mode displacement cycles of a hinged–hinged plate in vacuo

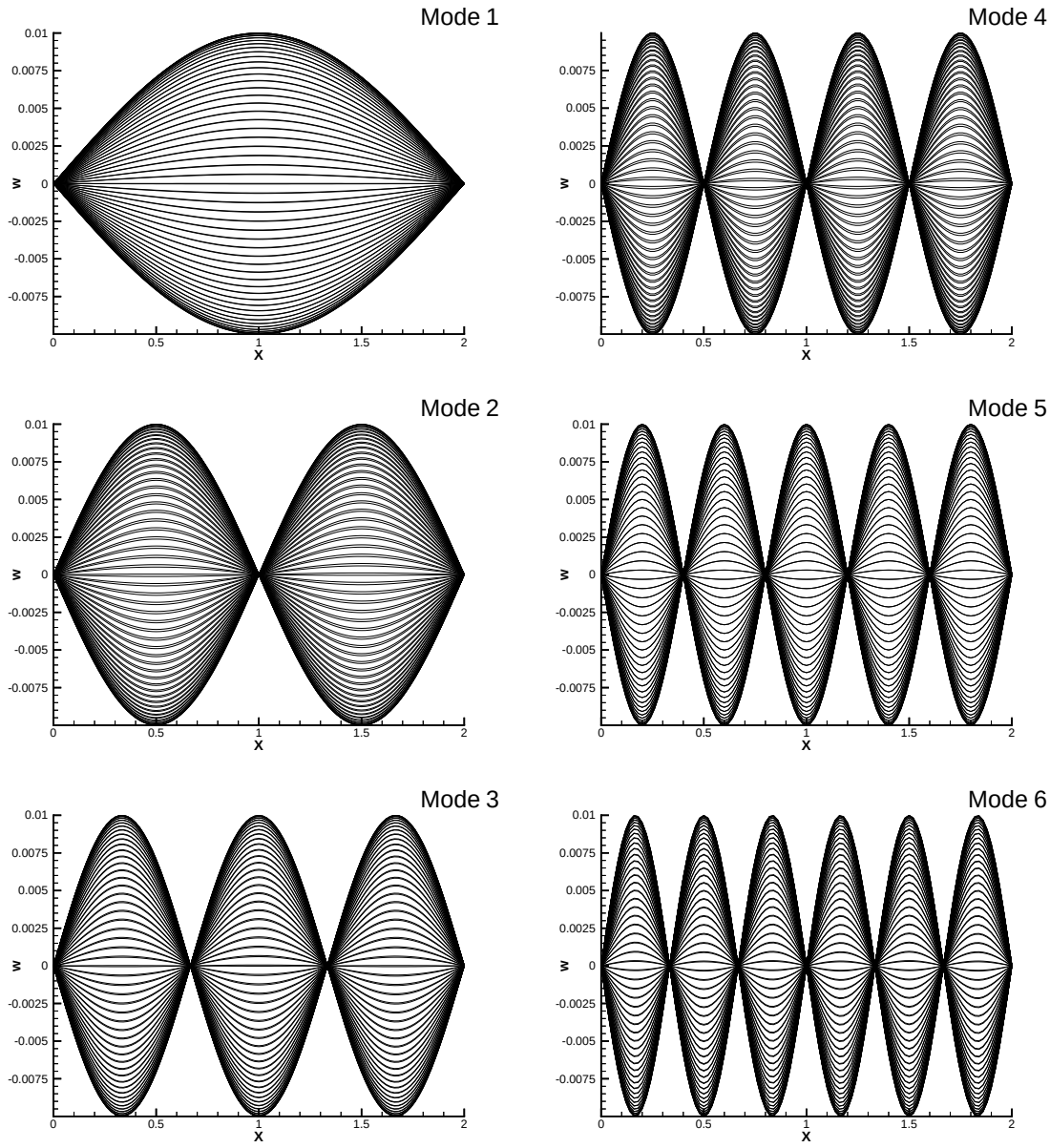


Figure 5.5: Mode 1 hinged–hinged oscillations under various conditions

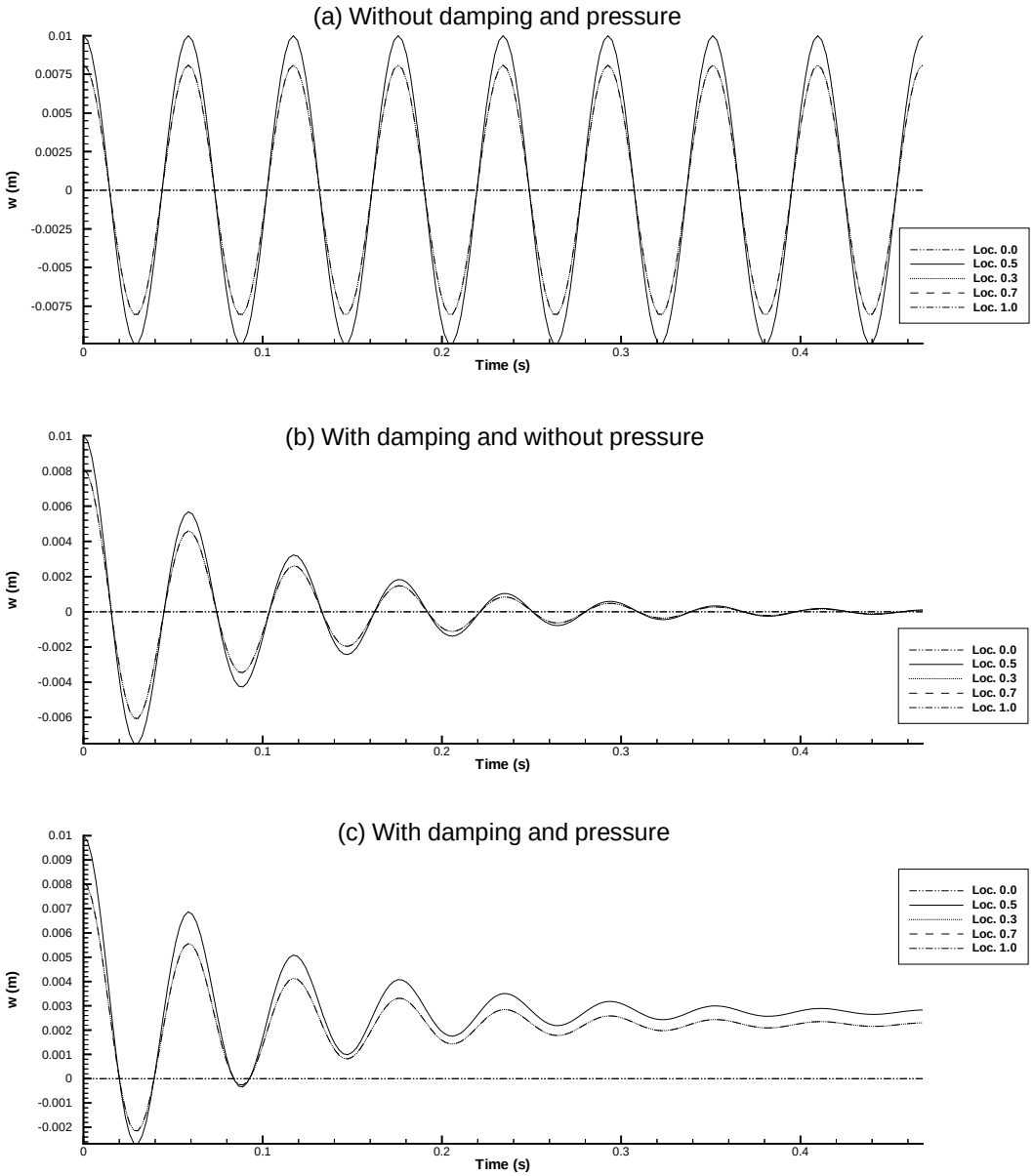


Figure 5.6: Mode 2 hinged–hinged oscillations under various conditions

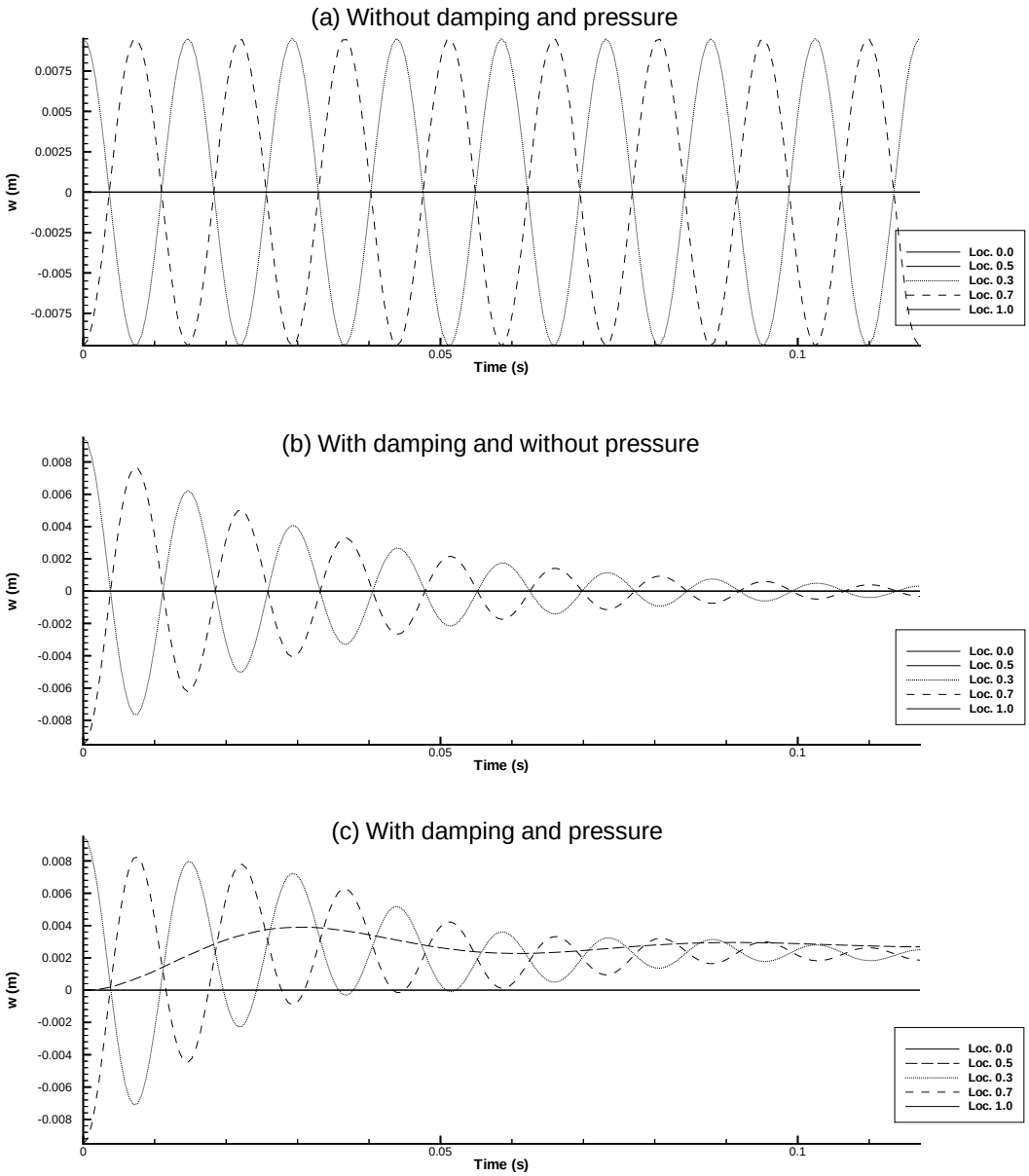


Figure 5.7: Mode 3 hinged–hinged oscillations under various conditions

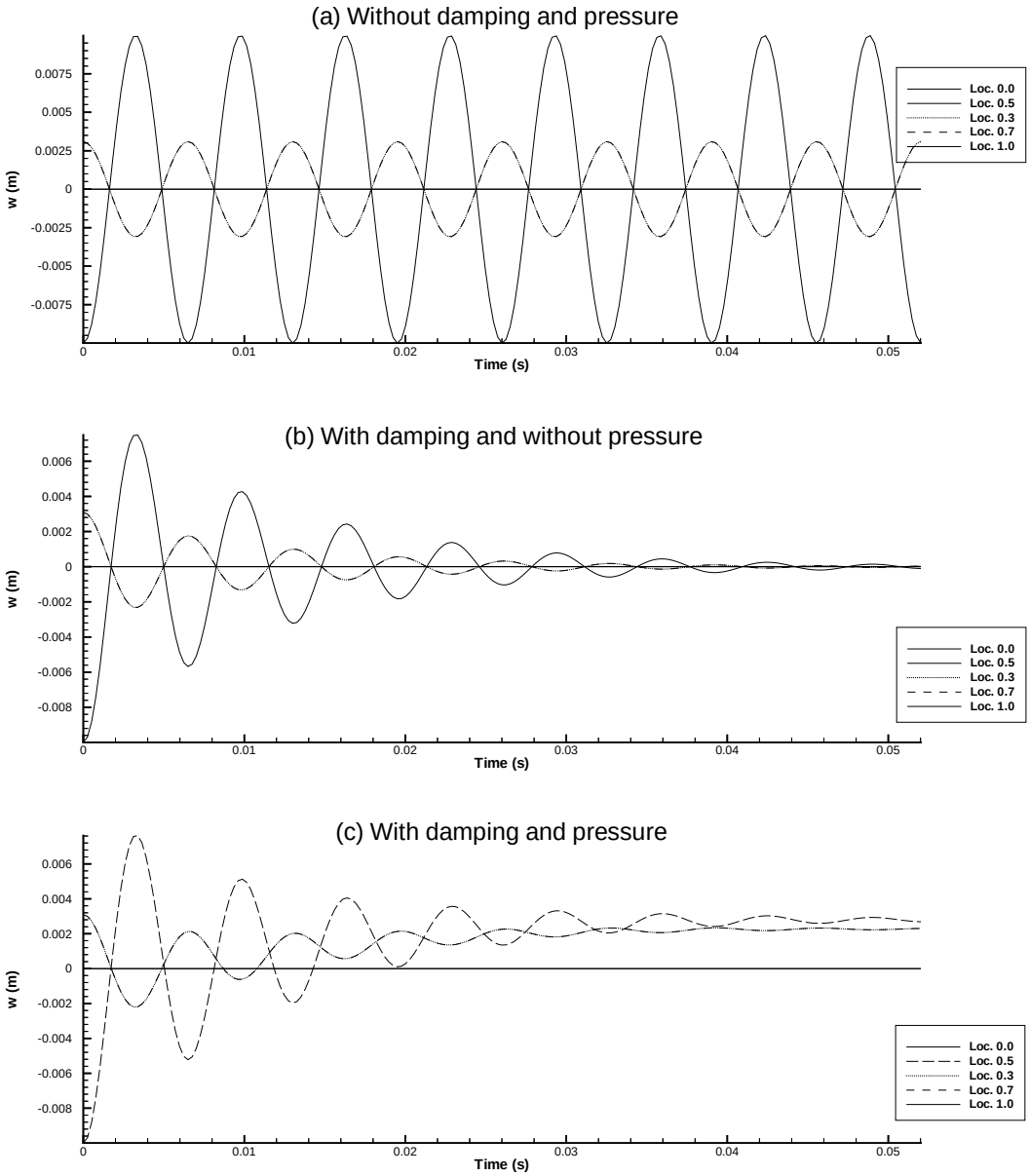


Table 5.4: Test matrix for hinged–hinged plates with pressure and damping

	$Case1$ $Mode1$	$Case2$ $Mode2$	$Case3$ $Mode3$
n	1	2	3
L	2	2	2
B	4921.86	4921.86	4921.86
d	50	150	450
p	-25	-25	-25
ω	107.35	429.41	966.18
$8 \times T$	0.4682	0.1170	0.0520
Δt	$1E-6$	$1E-6$	$1E-6$
$\#\ of\ sets$	200	200	200
$print\_step$	2341	585	260

5.2.3 Clamped–clamped plate fixed at both ends

The test matrices for the clamped–clamped configuration are given in Tables 5.5 and 5.6 for fundamental modes in vacuo, in Table 5.7 for in vacuo with damping and in Table 5.8 for conditions with pressure and damping. For all cases the elastic modulus, Poisson ratio, density and plate thickness were set to $E = 5.25 \times 10^{+13}\ N/m^2$, $\nu = 0.33333$, $\rho = 2600\ kg/m^3$ and $h = 0.001\ m$, respectively.

The first displacement cycle for all six eigenmodes are shown in Figure 5.8.

The explanation for Figures 5.9, 5.10 and 5.11 are virtually the same as for Figures 5.5, 5.6 and 5.7, respectively. The different end conditions result in different eigenvalues and eigenfunctions, but the general trend and behaviour of the oscillations are the same. For completeness the figures are described below.

Figure 5.9 gives the oscillations for Mode 1 (a) in vacuo without damping, (b) in vacuo with

Table 5.5: Test matrix for clamped–clamped plates in vacuo

	$Case1$ $Mode1$	$Case2$ $Mode2$	$Case3$ $Mode3$	$Case4$ $Mode4$	$Case5$ $Mode5$	$Case6$ $Mode6$
n	1	2	3	4	5	6
L	2	2	2	2	2	2
B	4921.86	4921.86	4921.86	4921.86	4921.86	4921.86
ω	243.36	670.83	1315.09	2173.92	3247.46	4535.70
$T = 2\pi/\omega$	0.0258	0.0093	0.0047	0.0029	0.0019	0.0014
Δt	$1E-6$	$1E-6$	$1E-6$	$1E-6$	$1E-6$	$1E-6$
$\#\ of\ sets$	100	100	100	100	100	100
$print\_step$	258	93	47	28	19	13

Table 5.6: Test matrix for clamped–clamped plates in vacuo (long runs)

	$Case1a$ $Mode1$	$Case2a$ $Mode2$	$Case3a$ $Mode3$
n	1	2	3
L	2	2	2
B	4921.86	4921.86	4921.86
ω	243.36	670.83	1315.09
$8 \times T$	0.2065	0.0749	0.0382
$\#\ of\ sets$	200	200	200
$print\_step$	1032	374	191

damping and (c) with pressure and damping. Figure 5.10 gives the oscillations for Mode 2 (a) in vacuo without damping, (b) in vacuo with damping and (c) with pressure and damping. Figure 5.11 gives the oscillations for Mode 3 (a) in vacuo without damping, (b) in vacuo with damping and (c) with pressure and damping.

5.2.4 Cantilevered plate fundamental modes

The test matrices for the cantilevered configuration are given in Tables 5.9 and 5.10 for fundamental modes in vacuo, in Table 5.11 for in vacuo with damping and in Table 5.12 for conditions

Table 5.7: Test matrix for clamped–clamped plates in vacuo with damping

	$Case1$ $Mode1$	$Case2$ $Mode2$	$Case3$ $Mode3$
n	1	2	3
L	2	2	2
B	4921.86	4921.86	4921.86
d	150	300	600
ω	243.36	670.83	1315.09
$8 \times T$	0.2065	0.0749	0.0382
$\#\ of\ sets$	200	200	200
$print\_step$	1032	374	191

Table 5.8: Test matrix for clamped–clamped plates with pressure and damping

	$Case1$ $Mode1$	$Case2$ $Mode2$	$Case3$ $Mode3$
n	1	2	3
L	2	2	2
B	4921.86	4921.86	4921.86
d	150	300	600
p	-6000	-6000	-6000
ω	243.36	670.83	1315.09
$8 \times T$	0.2065	0.0749	0.0382
$\#\ of\ sets$	200	200	200
$print\_step$	1032	374	191

Figure 5.8: The first six mode displacement cycles of a clamped–clamped plate in vacuo

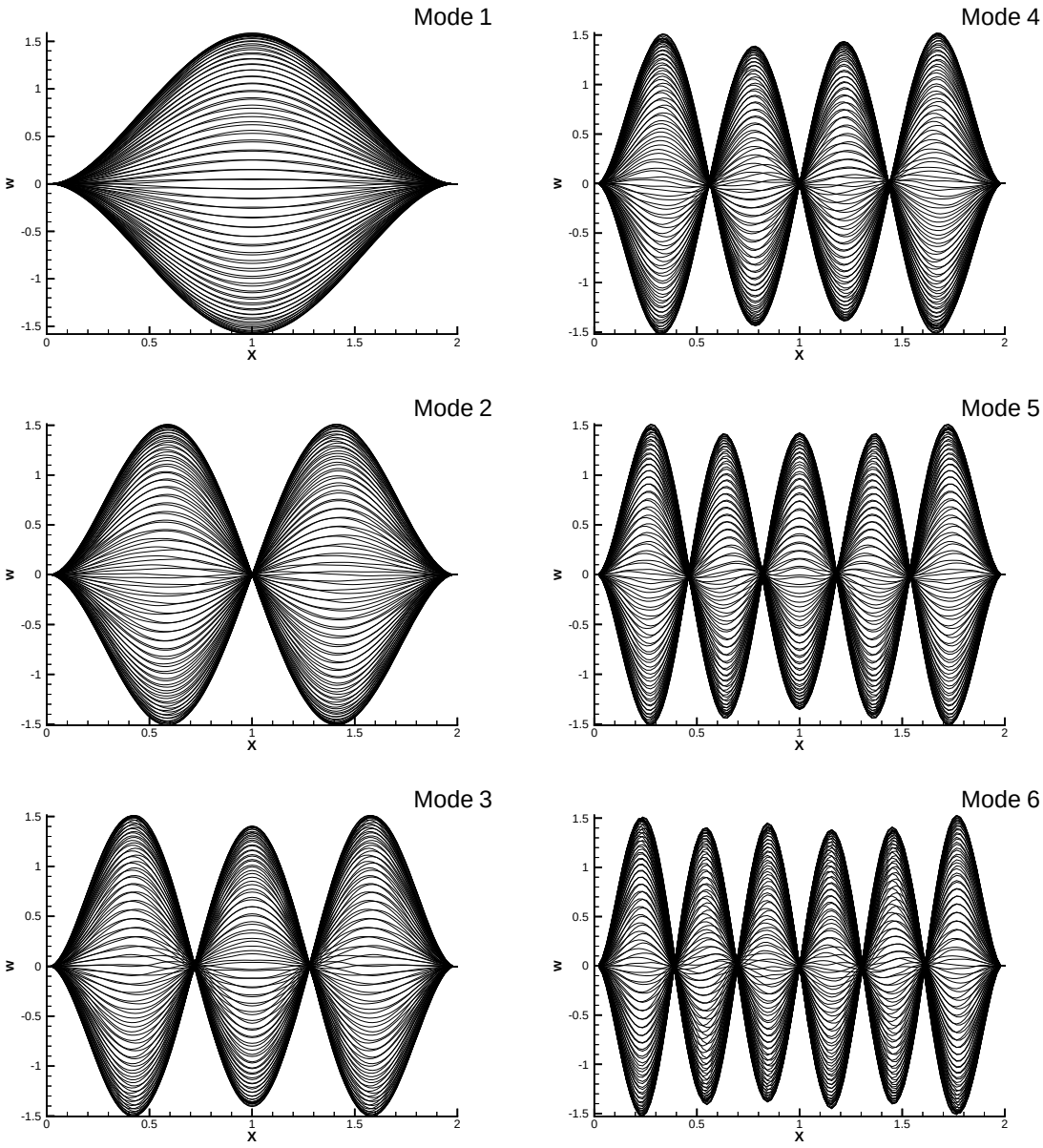


Figure 5.9: Mode 1 clamped–clamped oscillations under various conditions

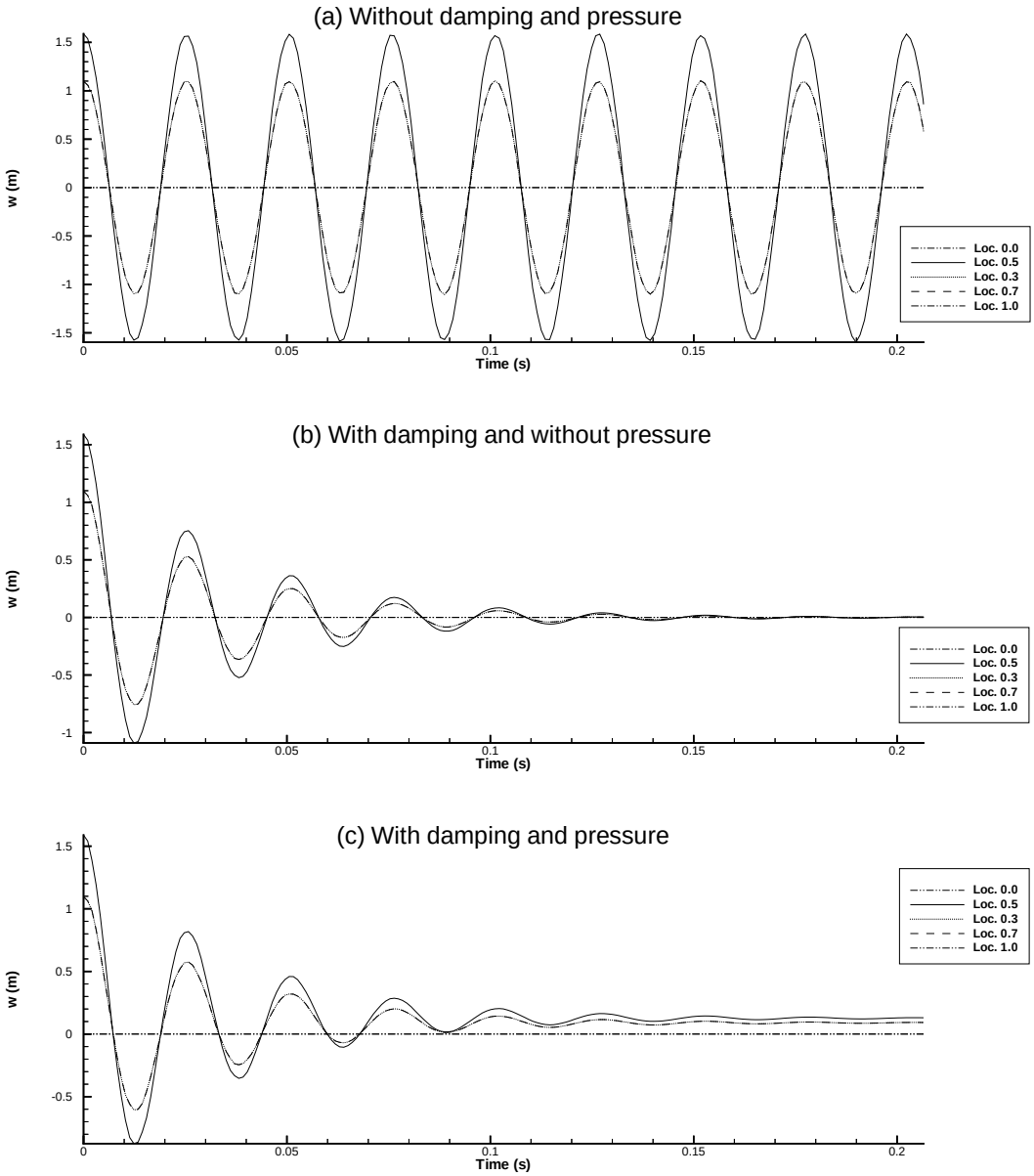


Figure 5.10: Mode 2 clamped–clamped oscillations under various conditions

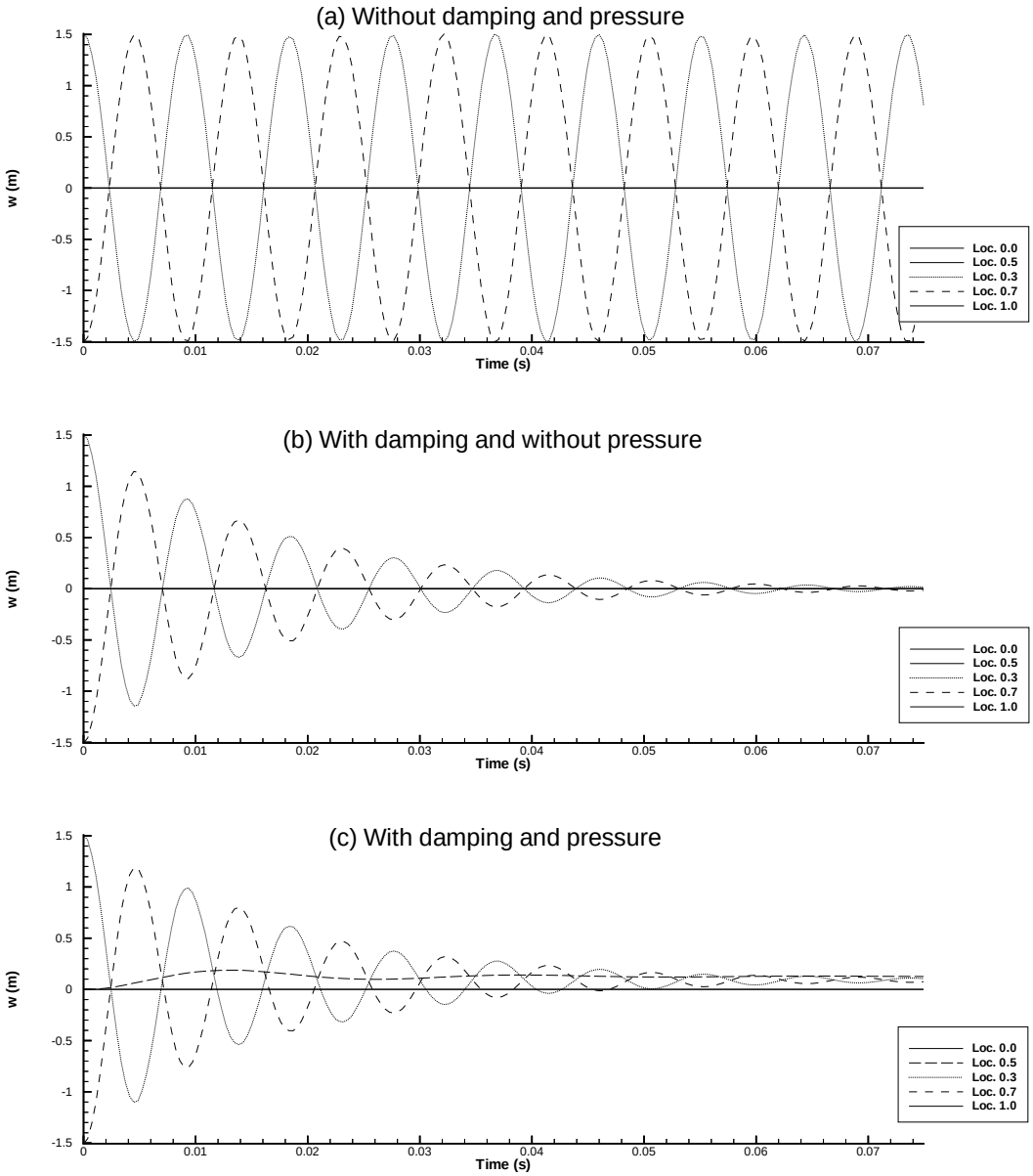
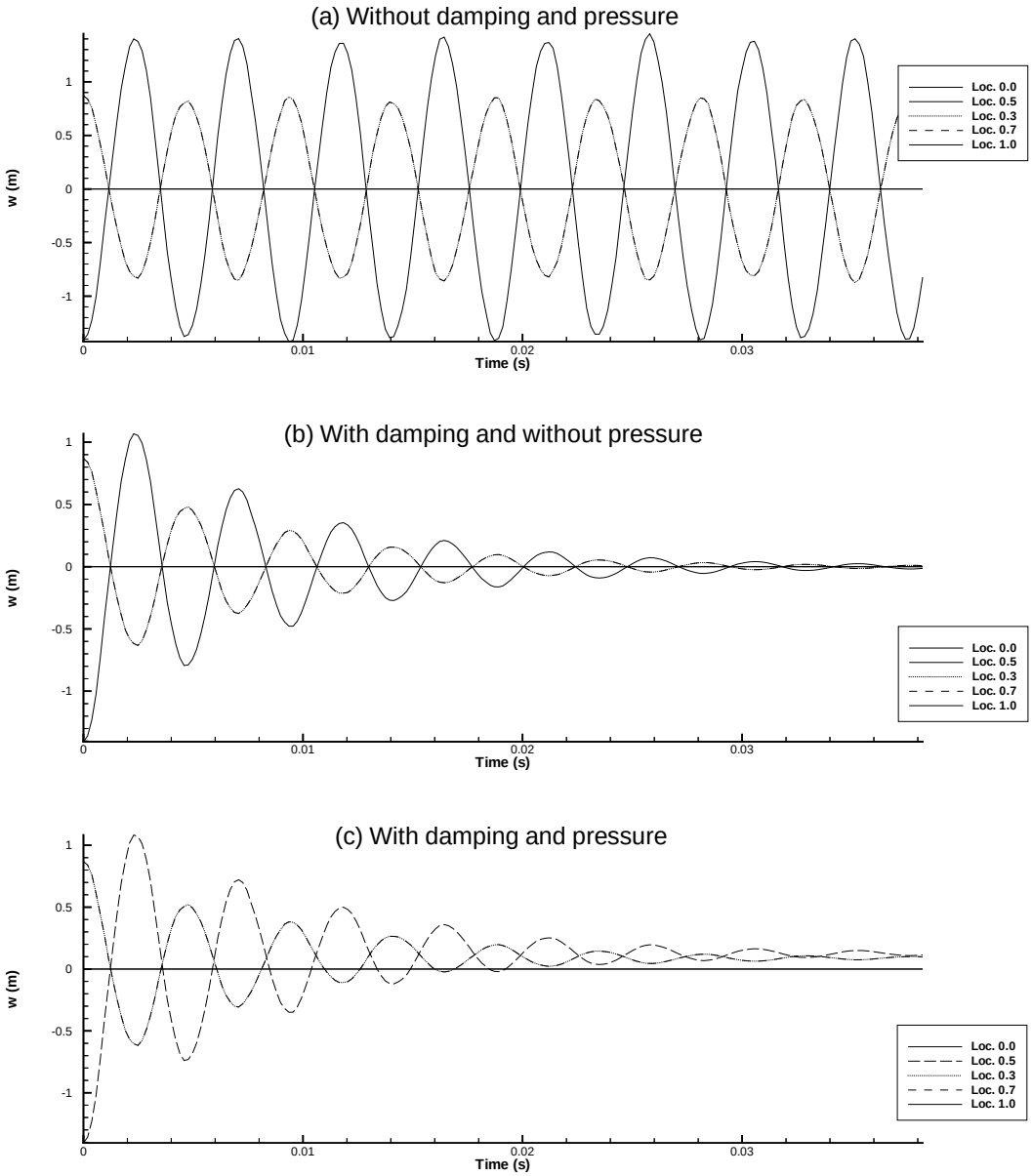


Figure 5.11: Mode 3 clamped–clamped oscillations under various conditions



with pressure and damping. For all cases the elastic modulus, Poisson ratio, density and plate thickness and scaling constant were set to $E = 5.25 \times 10^{+13}\ N/m^2$, $\nu = 0.33333$, $\rho = 2600\ kg/m^3$, $h = 0.001\ m$ and $C(const.) = 0.01$, respectively.

The first displacement cycle for all six eigenmodes are shown in Figure 5.12.

Figure 5.13 gives the oscillations for Mode 1 (a) in vacuo without damping, (b) in vacuo with damping and (c) with pressure and damping. Due to the geometry of the time series shown in Figure 5.12–Mode 1, the nodal displacement time series are in phase (see figure (a)). The same behaviour can be seen for the damped (b) and pressure and damped (c) cases. In vacuo the oscillations settle at zero displacement. When pressure is applied, the nodal points settle at an offset position from zero displacement.

Figure 5.14 gives the oscillations for Mode 2 (a) in vacuo without damping, (b) in vacuo with damping and (c) with pressure and damping. For mode two, the maximum initial displacement is at around Location 0.5, and the minimum displacement is at the end of the plate at Location 1.0. The first 4 monitored locations (i.e., 0.0, 0.3, 0.5 and 0.7) are in phase while Location 1.0 is 180^o out of phase, as shown in figures (a) to (c). When pressure and damping are applied, the plate exhibits a secondary wave motion, which eventually settles in an offset position where the pressure and the plate restoring forces are in equilibrium.

Figure 5.15 gives the oscillations for Mode 3 (a) in vacuo without damping, (b) in vacuo with damping and (c) with pressure and damping. The behaviour of the Mode 3 plate displacement time series is similar to the Mode 2 time series. It shows offset locations, where the same explanation applies as given for the previous figure.

Table 5.9: Test matrix for cantilevered plates in vacuo

	$Case1$ $Mode1$	$Case2$ $Mode2$	$Case3$ $Mode3$	$Case4$ $Mode4$	$Case5$ $Mode5$	$Case6$ $Mode6$
n	1	2	3	4	5	6
L	2	2	2	2	2	2
B	4921.86	4921.86	4921.86	4921.86	4921.86	4921.86
β	1.875	4.694	7.855	10.996	14.137	17.278
λ	0.9375	2.347	3.9275	5.498	7.0685	8.6393
ω	38.24	239.66	671.13	1315.18	2173.86	3247.45
$T = 2\pi/\omega$	0.1643	0.0262	0.0093	0.0048	0.0029	0.0019
Δt	$1E-6$	$1E-6$	$1E-6$	$1E-6$	$1E-6$	$1E-6$
$\#\ of\ sets$	100	100	100	100	100	100
$print\_step$	1643	262	93	47	28	19

Table 5.10: Test matrix for cantilvered plates in vacuo (long runs)

	$Case1a$ $Mode1$	$Case2a$ $Mode2$	$Case3a$ $Mode3$
n	1	2	3
L	2	2	2
B	4921.86	4921.86	4921.86
β	1.875	4.694	7.855
λ	0.9375	2.347	3.9275
ω	38.2402	239.6648	671.1359
$8 \times T$	1.3144	0.2097	0.0749
$\#\ of\ sets$	200	200	200
$print\_step$	6572	1048	374

Table 5.11: Test matrix for cantilevered plates in vacuo with damping

	$Case1$ $Mode1$	$Case2$ $Mode2$	$Case3$ $Mode3$
n	1	2	3
L	2	2	2
B	4921.86	4921.86	4921.86
d	25	50	100
β	1.875	4.694	7.855
λ	0.9375	2.347	3.9275
ω	38.24	239.66	671.13
$8 \times T$	1.3144	0.2097	0.0749
$\#\ of\ sets$	200	200	200
$print\_step$	6572	1048	374

Table 5.12: Test matrix for cantilevered plates with pressure and damping

	$Case1$ $Mode1$	$Case2$ $Mode2$	$Case3$ $Mode3$
n	1	2	3
L	2	2	2
B	4921.86	4921.86	4921.86
d	25	50	100
p	-5	-5	-5
β	1.875	4.694	7.855
λ	0.9375	2.347	3.9275
ω	38.24	239.66	671.13
$8 \times T$	1.3144	0.2097	0.0749
$\#\ of\ sets$	200	200	200
$print\_step$	6572	1048	374

Figure 5.12: The first six mode displacement cycles of a cantilevered plate in vacuo

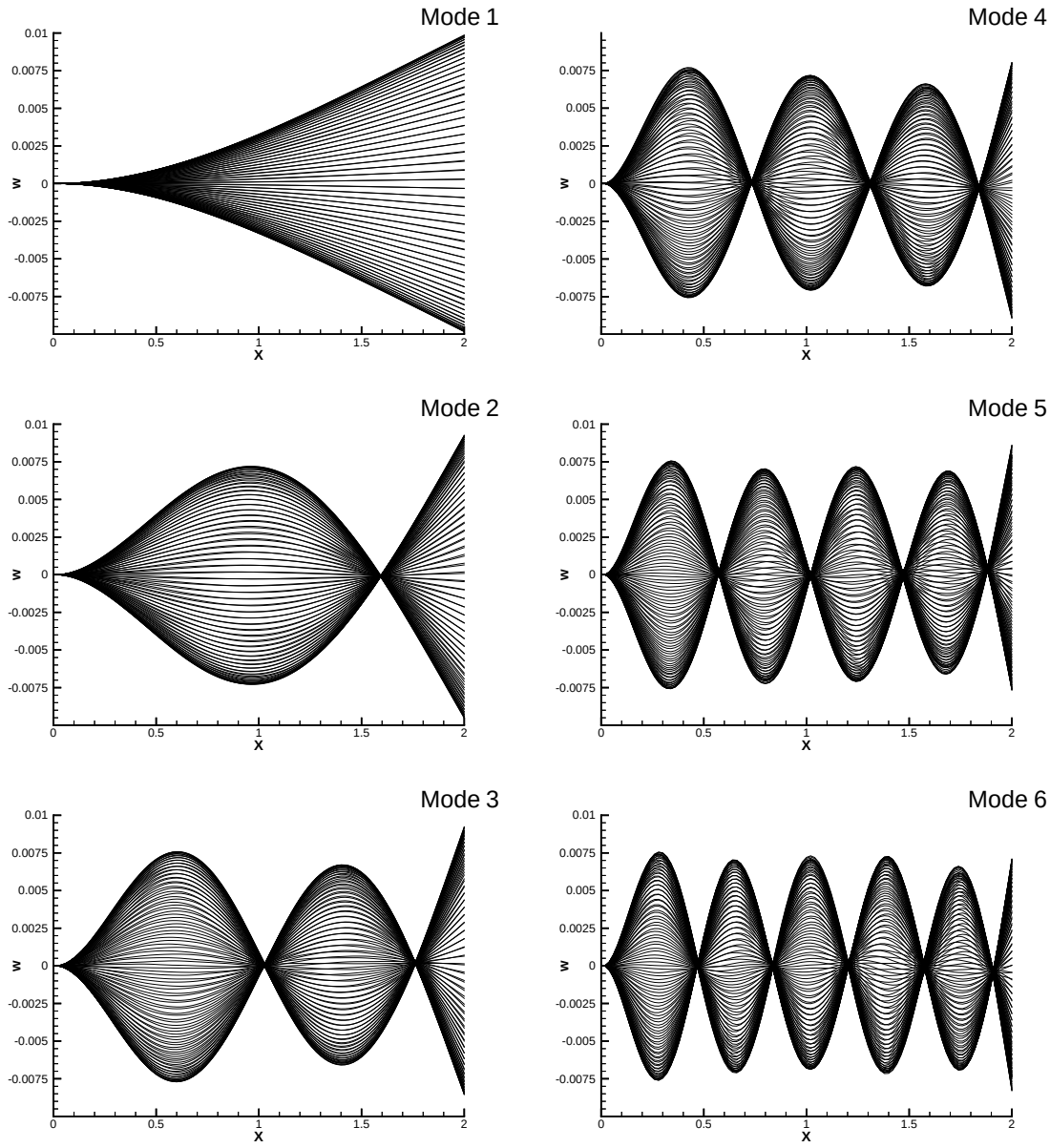


Figure 5.13: Mode 1 clamped–free cantilevered oscillations under various conditions

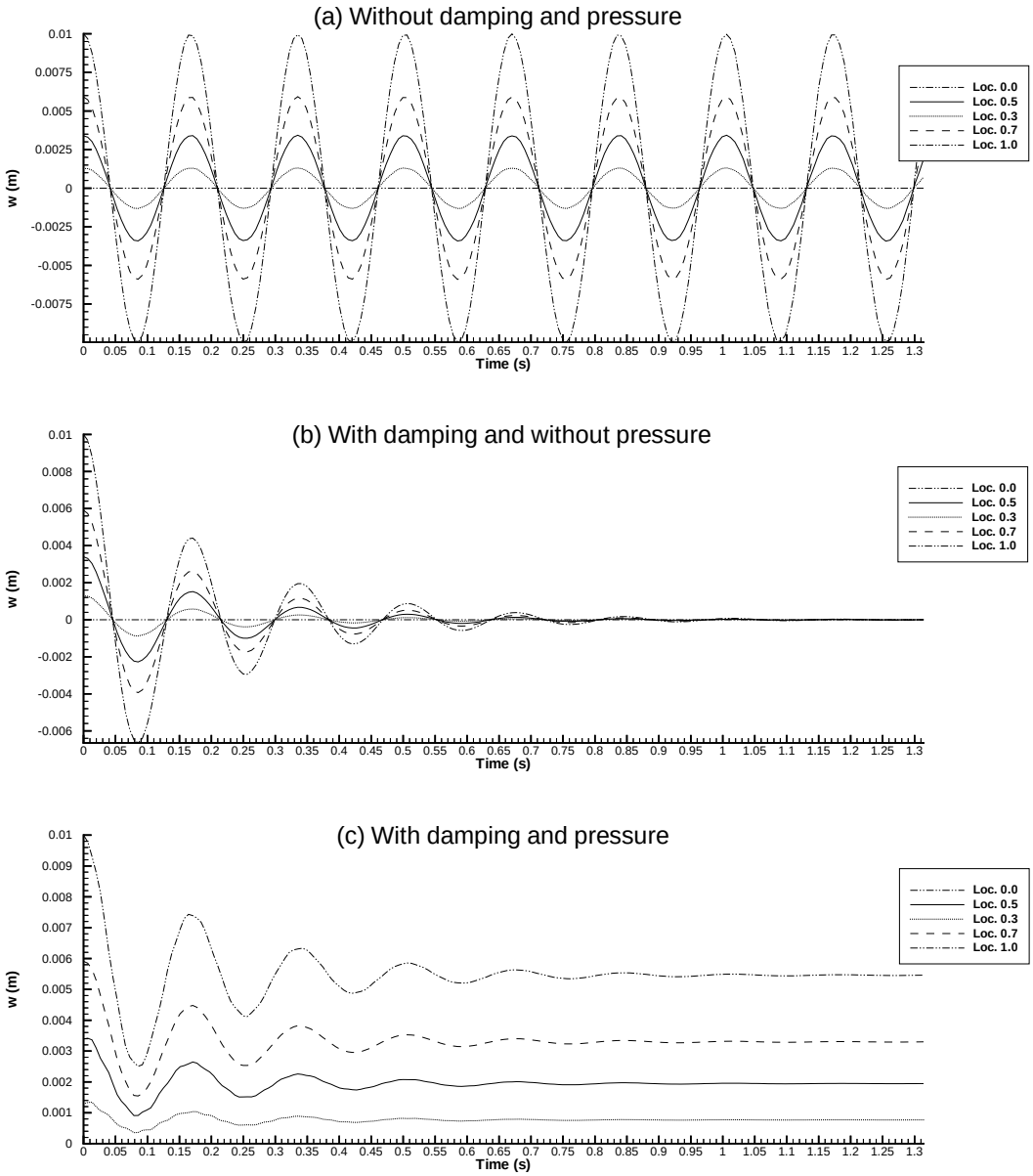


Figure 5.14: Mode 2 clamped–free cantilevered oscillations under various conditions

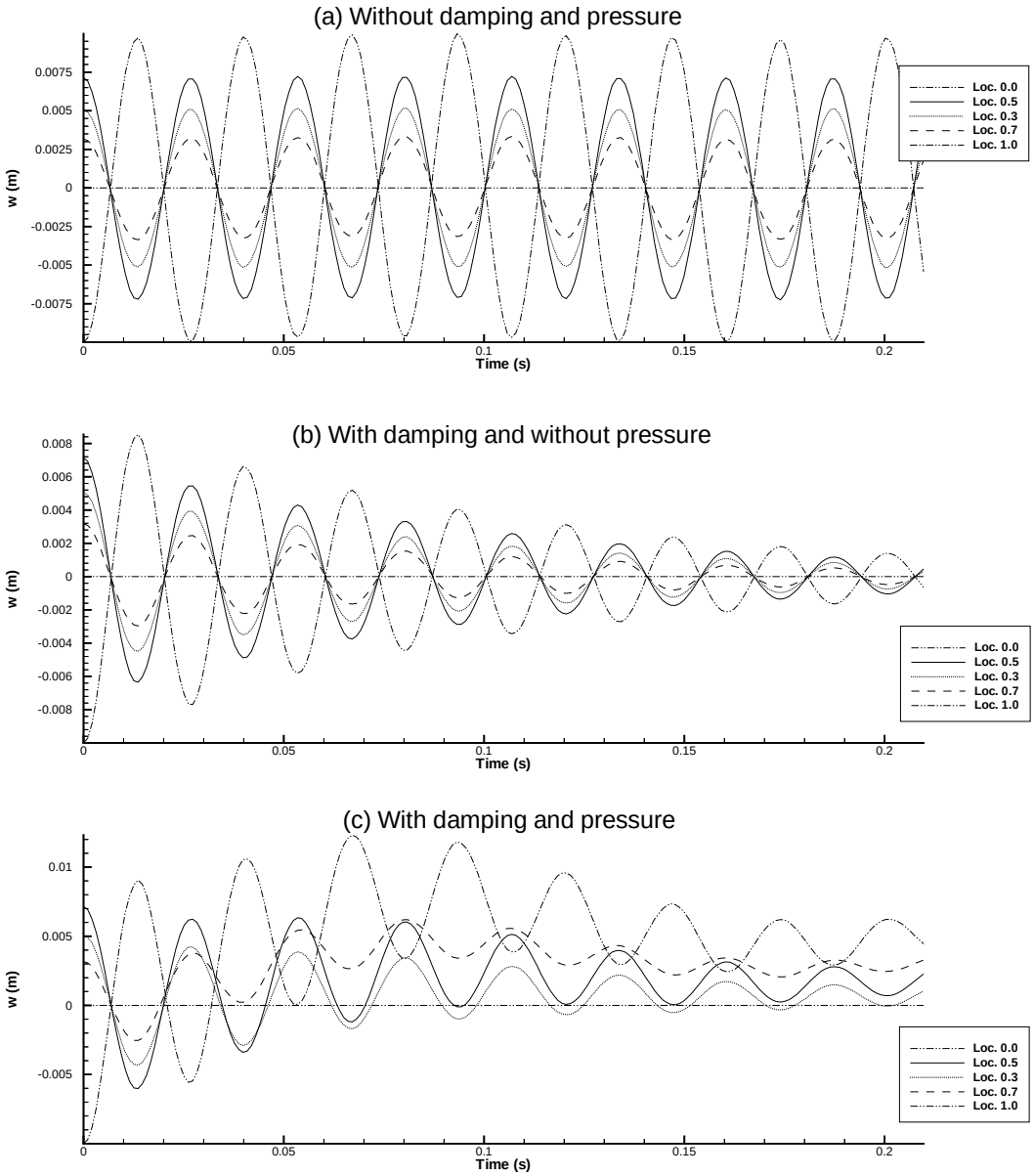
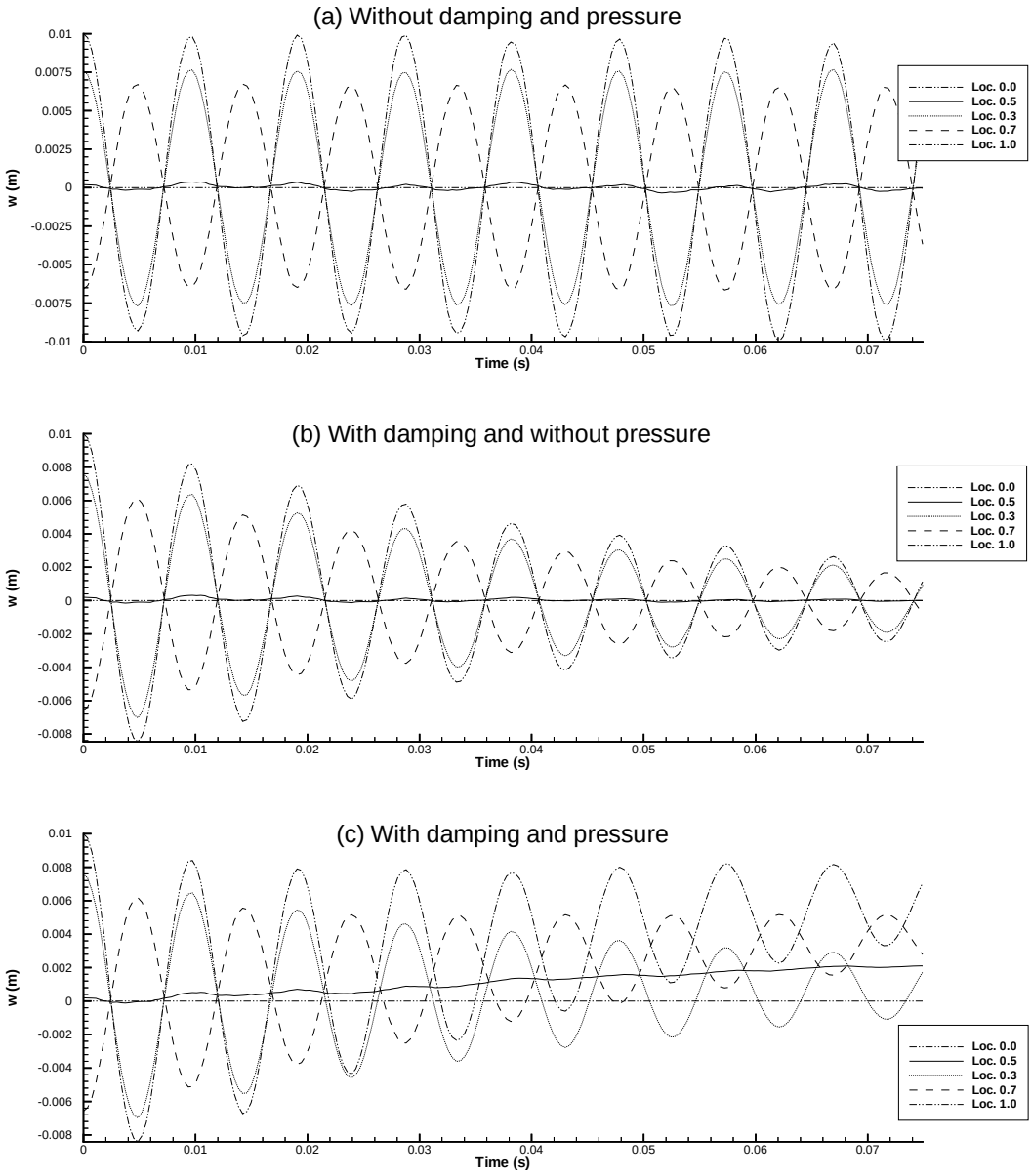


Figure 5.15: Mode 3 clamped–free cantilevered oscillations under various conditions



5.2.5 Static droop of a clamped cantilevered plate under gravity force

Following the validation of the code using fundamental modes, a specific initial shape of the flexible plate was chosen. One of the most natural choices is the curve of static droop of a plate with a clamped fixed end and a free cantilevered end shaped under the force of gravity alone. The initial condition at $t = 0$ has only the weight as a driving force, which causes the plate to "droop" downwards, following a curve [Hua95] defined by

$$w_g(x) = \frac{-\rho_m h g}{24B}(x^4 + 4Lx^3 + 6L^2x^2) \qquad x \in [0, L] \tag{5.33}$$

At the beginning of the time series the gravity is taken away, which started the oscillations of the plate.

The test matrix for the static droop under gravity force configuration is given in Table 5.13. For all cases the elastic modulus, Poisson ratio, density and plate thickness were set to $E = 5.25 \times 10^{+13}\ N/m^2$, $\nu = 0.33333$, $\rho = 3000\ kg/m^3$, $h = 0.001\ m$, respectively. The flexural rigidity was $B = 4921.86$.

Figure 5.16 gives the oscillations for (a) in vacuo without damping, (b) in vacuo with damping and (c) with pressure and damping. The static droop initial curve is not one of the fundamental mode curves. Therefore, the time series exhibits traveling waves, which makes the time series "noisy". The initial displacement is similar to a Mode 1 displacement curve shown as the outline of Figure 5.12–Mode 1 and the behaviour is the same as explained for Figure 5.13.

5.2.6 Conclusions

The computer code, developed for the modeling of the linear transverse displacement of a flexible plate, was successfully verified and tested. Code verification was performed using the FORTRAN compiler's Debugger, by stepping through the code and checking the variables as they were calculated. Code validation was performed by comparing predictions to analytical results. The code was tested for the first six eigenmodes under the following configurations:

Figure 5.16: Static droop of a clamped cantilevered plate under various conditions

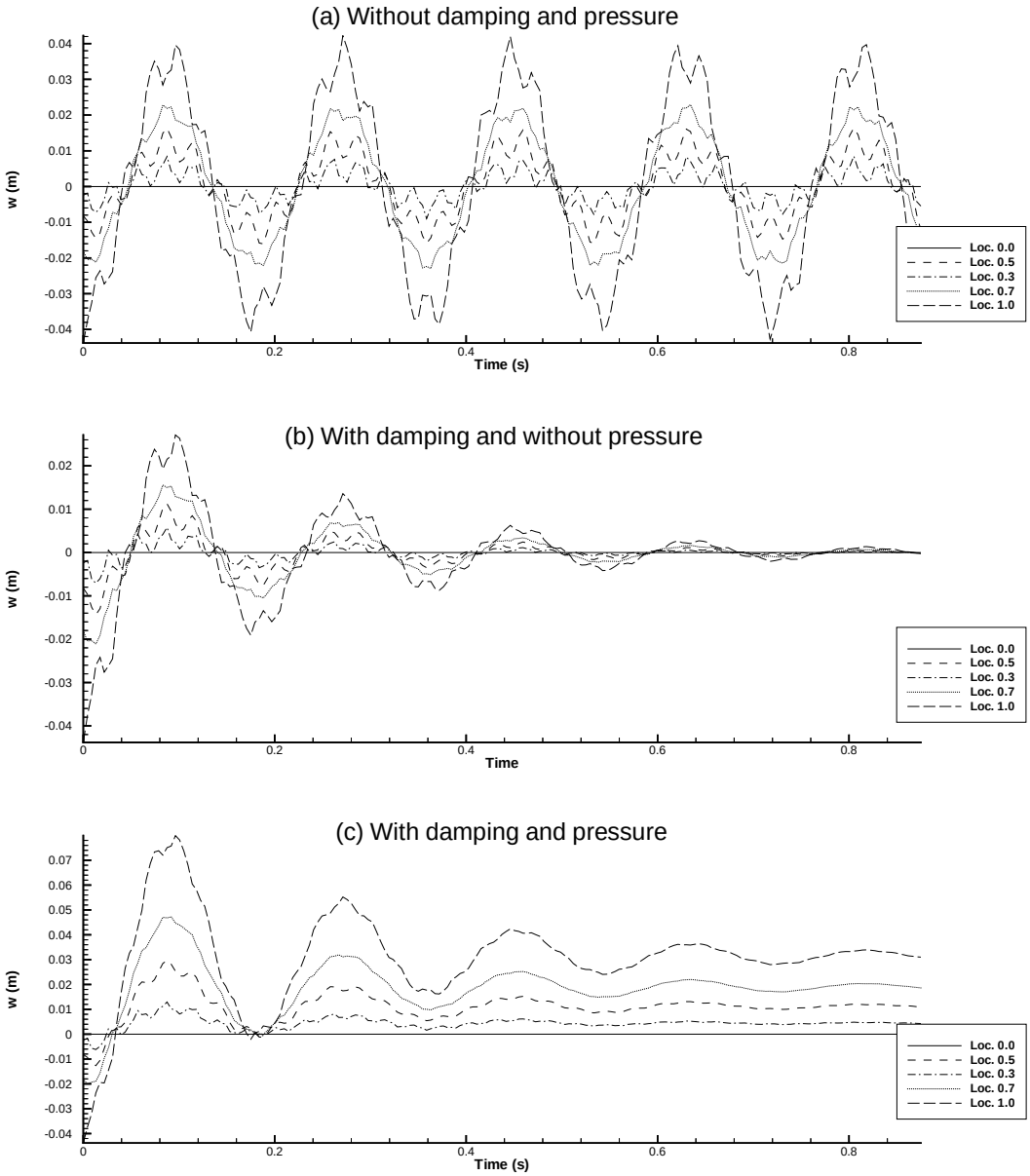


Table 5.13: Test matrix for the static droop under gravity cases

	*Case*1 1 *Cycle*	*Case*2 *Long Run*	*Case*3 *Damping*	*Case*4 *Pressure & Damping*
L	2	2	2	2
d	–	–	25	25
p	–	–	–	-25
Δt	$1E-6$	$1E-6$	$1E-6$	$1E-6$
$T(ad\ hoc)$	0.175	0.875	0.875	0.875
$\#\ of\ sets$	200	200	200	200
$print\_step$	875	4375	4375	4375

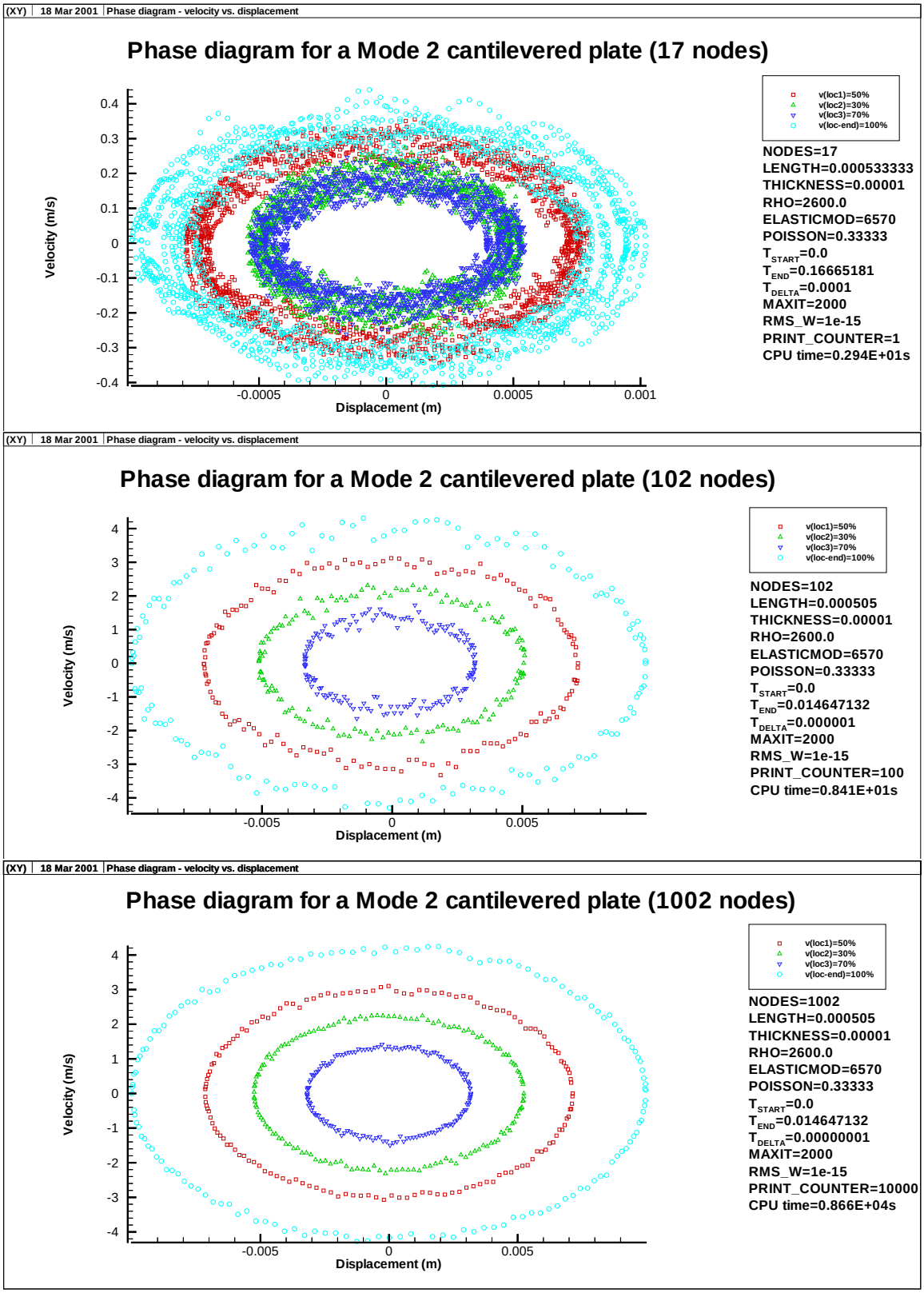
- Hinged–hinged plate fixed at both ends
- Clamped–clamped plate fixed at both ends
- Cantilevered plate with one end clamped and the other end free

In addition, the program was designed to calculate damping and changing pressure conditions.

It was found that the predictions accurately matched the displacement phase of the analytical solutions for all of the examined cases following a given number of time steps. In consequence, the code can be further developed as part of the final stage of the research, where it will be coupled with the FEM Navier–Stokes fluid code developed earlier.

For the input parameters and discretization used, the displacement oscillations were smooth. The corresponding velocity and acceleration values were more irregular. The effect of discretization on the displacement and velocity of a Mode 2 centilevered plate are shown in three phase diagrams, where the velocity is plotted against the nodal displacements (see Figure 5.17). The accuracy of velocity and acceleration results can be increased by reducing the RMS error criterion, by refining the mesh, by reducing the time step, or using a higher order numerical scheme.

Figure 5.17: Phase diagrams at various discretizations



Chapter 6

Fluid–structure interaction using coupled codes

Analysis of the data – provided by numerical experiments – is carried out in a number of ways. The displacements of the moving cantilevered plate at each nodal point are monitored during the time series simulation. Wall displacement profiles are shown in animations as a visual impression of the cantilevered plate movement. The velocity and pressure values at each nodal point are saved at predefined time steps. This allows for the animation of the $U1$, $U2$ and p values. Phase diagrams, where the time series velocity values are plotted against the displacement values for specified nodal locations, provide insight about the plate dynamics. Transmural surface pressure time series between the upper and lower nodal points of the cantilevered plate are also animated. Further understanding of the plate dynamics are gained from the flexible plate energy evaluations.

After the completed validation exercises for the fluid and compliant wall codes, the next intermediate step towards coupling of these code components was their partial coupling, in which a static displacement of the cantilevered plate was prescribed as the internal domain boundary. The results from this analysis are presented in Appendix O. The code captured the expected pressure distribution along the displaced plate. Therefore, it was concluded that the coupling of the fluid and wall codes will yield reasonable results for the dynamic interaction between

the fluid flow and the compliant surfaces.

Aeroelastic (flutter) analysis examines the fluid dynamic instabilities, such as flutter and divergence, to gain better understanding of the phenomena.

Divergence is an aeroelastic phenomenon where the aerodynamic forces are greater than the elastic restoring forces of the structure. The aerodynamic forces are a function of the flow velocity while the elastic forces are independent of flow velocity, so there is a velocity above which a small deformation due to the aerodynamic forces will lead to a large deflection in the structure.

Flutter is a dynamic instability that involves coupling of fluid dynamic forces and elastic and inertial forces of the structure. As the flow velocity increases around the flexible wall section, the fluid dynamic forces interact with and alter the structural vibrations of the flexible wall.

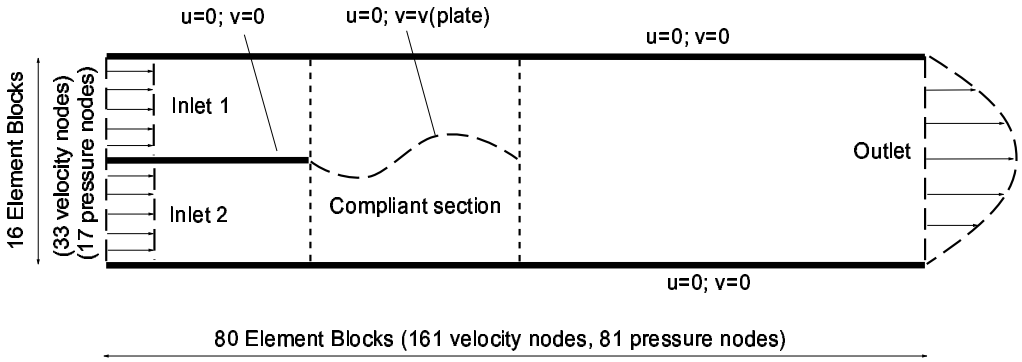
In a flow field an oscillating flexible wall generates unsteady fluid dynamic forces. The fluid loading and the wall response can couple and can cause phase shifts between the motion of the wall and the pressure acting on the wall. The velocity and the pressure of the flow affects the work done on the flexible wall as energy is extracted from the flow stream. At a critical flow speed, the work done on the plate equals the wall damping and the plate movement neither amplifies or dissipates. At speeds greater than a critical velocity, the energy extracted from the flow amplifies the plate movement resulting in flutter or divergence.

6.1 Coupled 2D duct flow/compliant cantilevered plate

6.1.1 Methodology and assumptions

A simplified model of the upper airway – a 2D duct – with a single flexible section, corresponding to the soft palate, is shown in Figure 6.1.

Figure 6.1: Split inlet model with compliant cantilevered section



The typical snoring frequency is between $30Hz$ and $100Hz$ [GJ93]. In these calculations the upper limit was used, for practical considerations to maximize the number of oscillations for given time steps. A $100Hz$ soft palate flutter frequency results in a 0.01 second cycle for each oscillation. Flow and wall stability calculations allowed for 5×10^{-5} second time steps for both, fluid and wall code components. Numerical simulations were performed for 5 cycles, resulting in a 0.05 second time series time. In the calculations only a flexible cantilevered section, corresponding to the soft palate, was considered. The transverse vibrations of the soft palate are far faster than the motion of the surrounding walls. For example, the pharynx has typical oscillation times of 1 to 2 per seconds. As a result, during snoring, the structural behaviour of the upper airway can be modeled as one of a flexible structure in a rigid channel [AD95]. (A conceptual model which includes compliant sections for the pharyngeal region will be given in Section 6.4.)

For the fluid code the fluid (air) density was set to $\rho = 1.1774 kg/m^3$, based on $T = 300K$ air temperature. The corresponding dynamic and kinematic viscosities were $\mu = 1.98 \times 10^{-5} kg/ms$ and $\nu = 1.68 \times 10^{-5} m^2/s$, respectively.

The inlet velocity was calculated from the airway flow area and the volumetric flow rate, up

to $\dot{V} = 0.0004m^3/s$. The three inlet flow velocities were $v_1 = 0.31831m/s$, $v_2 = 0.63662m/s$ and $v_3 = 1.27324m/s$. The highest velocity (v_3) was chosen based on practical considerations related to flow physics and code execution time. When the same fluid is used the Reynolds-number increases with the increase of the inlet velocity. The present numerical formulation does not include a turbulence model. Therefore, the prediction accuracy reduces as the Reynolds-number approaches the laminar-to-turbulent transition region. In addition, numerical stability is affected by the Reynolds-number. With the increase of Re, the time step must be reduced which increases code execution time. As shown in Table 4.22 at Re=1512 a Δt between $4.4 \times 10^{-5}s$ and $8.9 \times 10^{-5}s$ is recommended. For the coupled split inlet cases $\Delta t = 5 \times 10^{-5}s$ is used. Additional inlet velocities were chosen to cover two more points in the test matrix, with values at one half and at one quarter of v_3.

The Reynolds numbers were calculated from the hydraulic diameters, inlet velocities and the kinematic viscosity, resulting in $Re = 378$, $Re = 756$ and $Re = 1512$. For the cases, where the lower inlet was blocked, the hydraulic diameter was reduced by 50%. Therefore, for those cases the Reynolds numbers were also reduced by 50%.

For the wall code the elastic modulus was set to $E = 8.8 \times 10^8 N/m^2$. The Poisson ratio was $\nu = 0.3333$, the plate density was $\rho_{wall} = 2477kg/m^3$, with a plate thickness of $h = 0.00001m$. This configuration, with the appropriate plate length, resulted in the desired $100Hz$ frequency – in vacuo.

6.1.2 Results and discussion

Two sets of numerical experiments were performed. The first set with two inlets and the second set with the lower inlet blocked. The test matrices for these are given in Tables 6.1 and 6.2, respectively.

For the two inlets cases, a typical example for a single cycle of oscillation is given in Figures 6.2,

Table 6.1: Test matrix for coupled code – two-inlet cases

Max. Displ. ($Mode$ 2)	$\mathrm{Re}_1 = 378$ $v_{inlet} = 0.32m/s$	$\mathrm{Re}_2 = 756$ $v_{inlet} = 0.64m/s$	$\mathrm{Re}_3 = 1512$ $v_{inlet} = 1.27m/s$
$w_1 = 0.0005$	$Case$ 1	$Case$ 2	$Case$ 3
$w_2 = 0.001$	$Case$ 6	$Case$ 7	$Case$ 8
$w_3 = 0.002$	$Case$ 11	$Case$ 12	$Case$ 13

Table 6.2: Test matrix for coupled code – one-inlet cases

Max. Displ. ($Mode$ 2)	$\mathrm{Re}_1 = 378$ $v_{inlet} = 0.64m/s$	$\mathrm{Re}_2 = 756$ $v_{inlet} = 1.27m/s$	$\mathrm{Re}_2 = 756$ $v_{inlet} = 0.64m/s$	$\mathrm{Re}_3 = 1512$ $v_{inlet} = 1.27m/s$
$w_1 = 0.0005$	$Case$ 4	$Case$ 5	$Case$ 4a	$Case$ 5a
$w_2 = 0.001$	$Case$ 9	$Case$ 10	–	–
$w_3 = 0.002$	$Case$ 14	$Case$ 15	$Case$ 14a	$Case$ 15a

6.3 and 6.4. The figures represent the $U1$ velocity components only. Once a quasi-steady-state condition is reached, the flexible plate is released from its Mode 2 initial displacement. While the plate blocks both channels, the blockage ratio is higher in the bottom channel than in the top channel, resulting in the highest local velocities around the end tip of the cantilevered plate. Due to the no-slip boundary conditions the velocities are zero at the walls. The plate curvature affects the fluid velocity in the vicinity of the wall, in the form of a "low velocity bubble". As the plate moves, this bubble shifts along the plate, while the flow also slows down behind the newly developed curvatures. The pressure distribution change is not in phase with the wall movement. This phase shift is the basis for a flutter condition, where the flow feeds energy into the plate, resulting in an instability.

For a duct with two inlets, the time series were calculated for three Reynolds-numbers at three different initial maximum wall displacements. For each case five cycles of oscillations

Figure 6.2: Coupled time series (1-3), 2D duct, $Re = 1512$, $v_{in} = 1.27m/s$, $w_{max} = 0.002m$ ((a) $t = 0s$, (b) $t = 0.0015s$, (c) $t = 0.003s$)

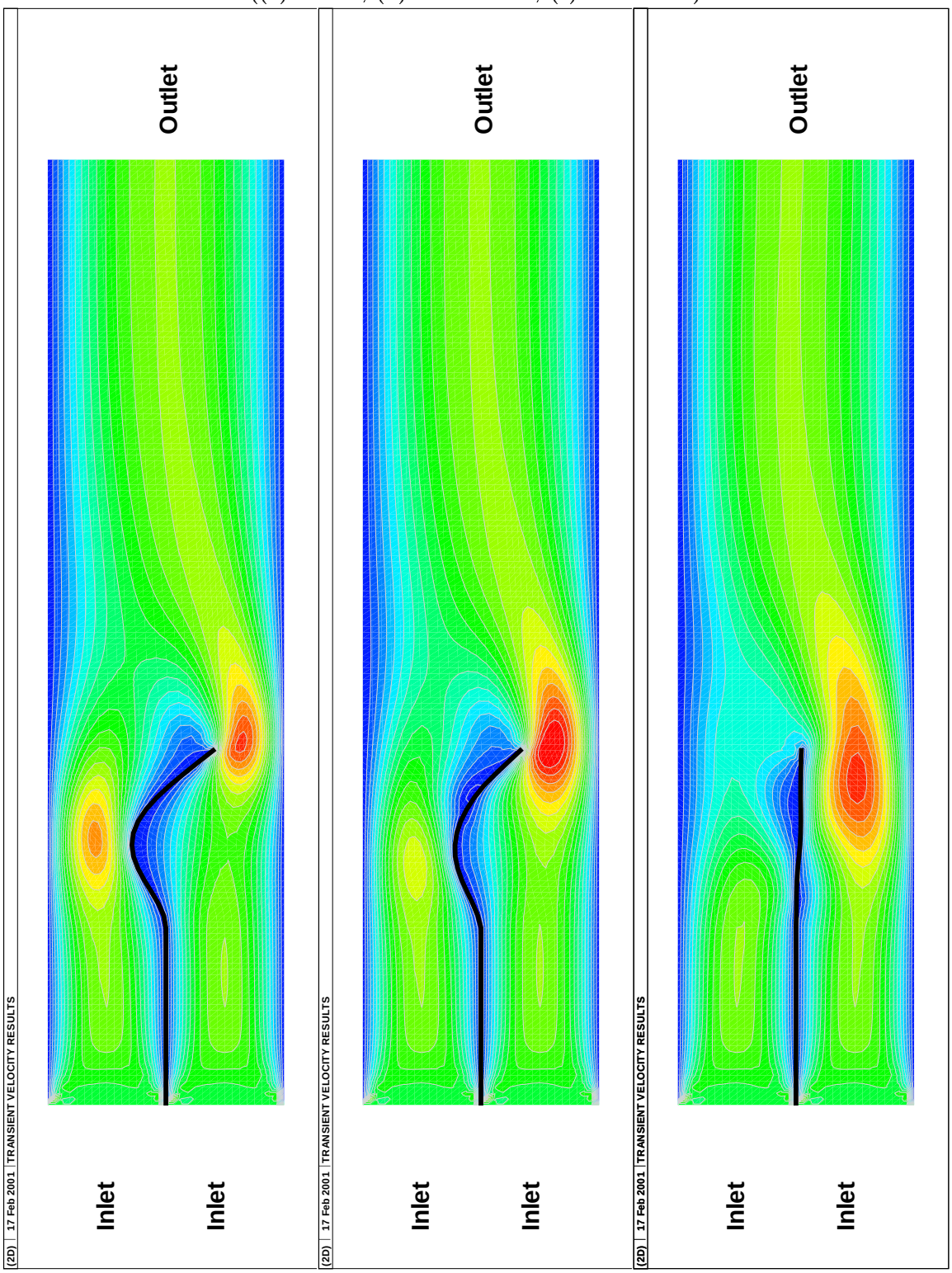


Figure 6.3: Coupled time series (4-6), 2D duct, $Re = 1512$, $v_{in} = 1.27m/s$, $w_{max} = 0.002m$ ((d) $t = 0.0045s$, (e) $t = 0.0055s$, (f) $t = 0.0065s$)

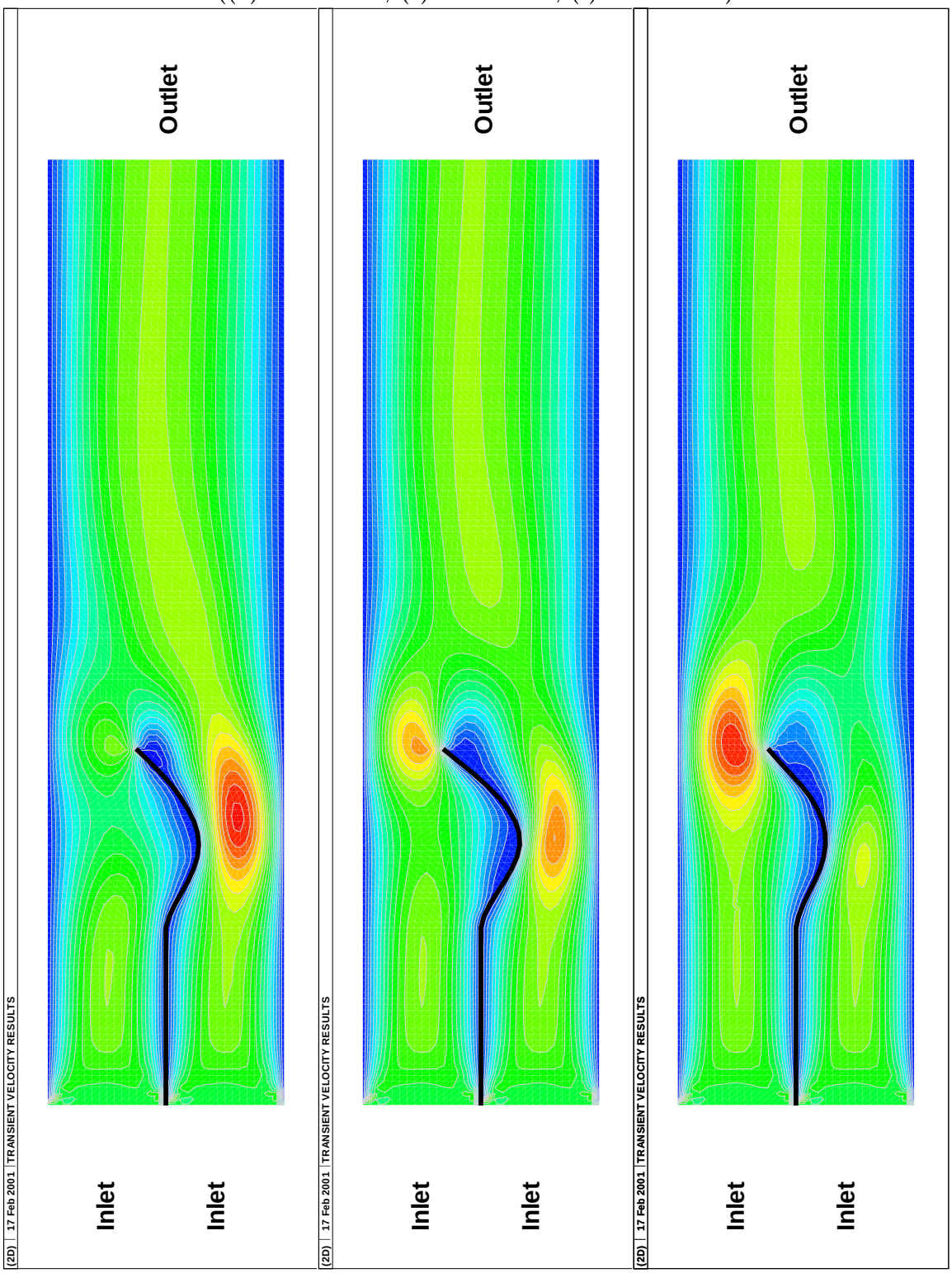
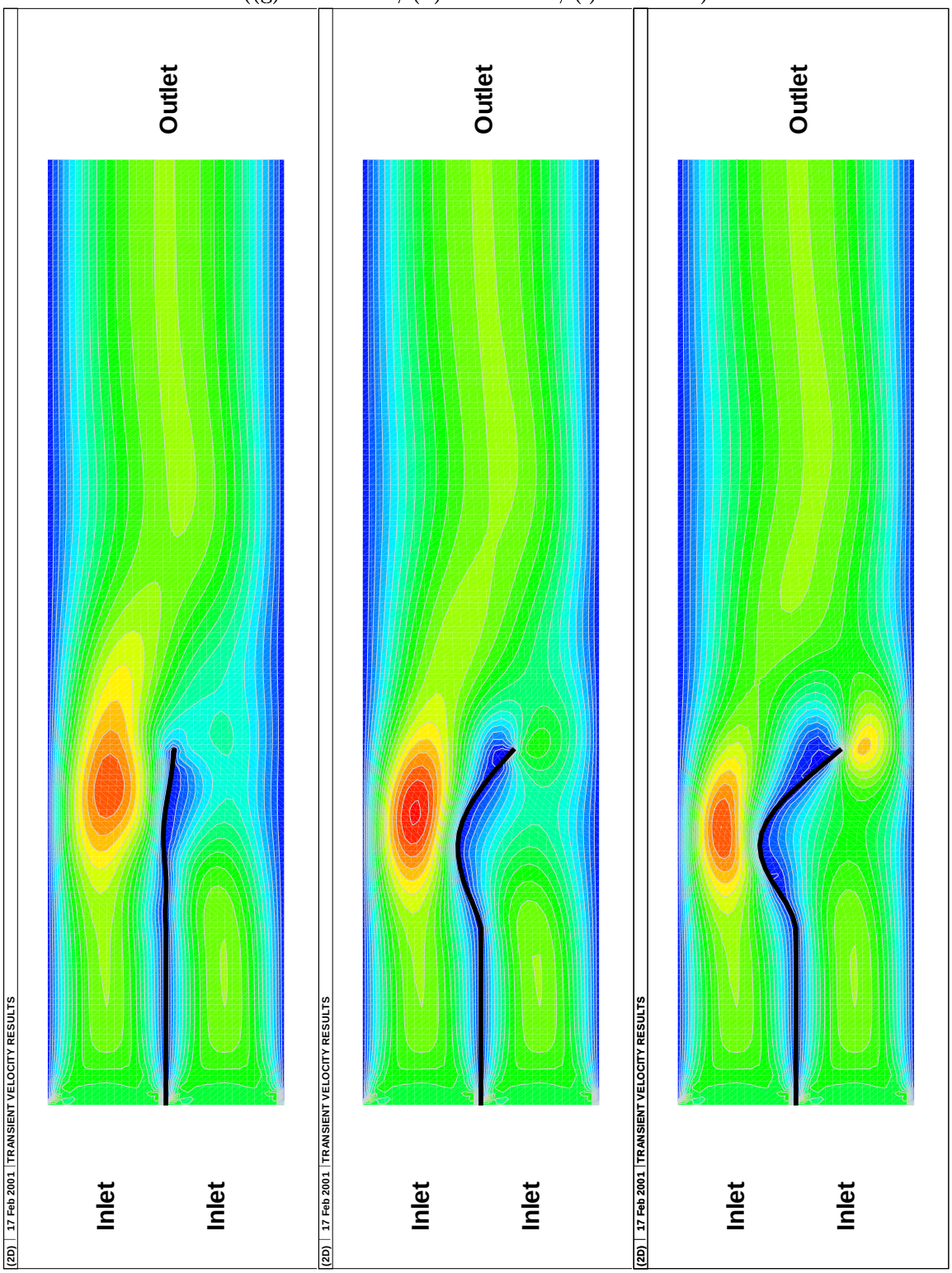


Figure 6.4: Coupled time series (7-9), 2D duct, $Re = 1512$, $v_{in} = 1.27m/s$, $w_{max} = 0.002m$

((g) $t = 0.008s$, (h) $t = 0.009s$, (i) $t = 0.01s$)



were modeled. (Each run required close to two days of CPU time on a Pentium III, 800 MHz computer.) The plate related variables were monitored at five pre-defined locations, at the beginning, end, middle and 30% and 70% from the beginning of the compliant section. The plate movement is demonstrated by plotting movement of these nodes, as shown in Figure 6.5. (Note that only the low (Re=378) and high (Re=1512) Reynolds-number cases are given, for each of the three maximum initial displacement cases.) At low Reynolds-numbers the plate oscillates in a regular orderly fashion, similar to those in vacuo. At higher Reynolds-numbers a second, slower harmonic seems to appear, but due to the relatively short time series its long term trend can not be determined. The same conclusion can also be derived from Figure 6.6, where the full plate movement is shown throughout the full time series. The starting position is highlighted in red and the end position in green. At low Reynolds-numbers ($Re = 378$) the nodal point of the plate corresponding to the location where the x-axis intersects the flexible plate is almost stationary and the plate displacement phase after the fifth oscillation is virtually the same as that of the starting position. The plate oscillation is very similar to in vacuo oscillations indicating minimal interaction between the flow and the plate. The slight movement of this stationary point is due to errors related to coarse plate discretization. The same behaviour was observed for the in vacuo cases as well. It was proven that with the increase of mesh size the accuracy of the nodal displacement increases in the time series and this slight swaying disappears. However, at higher flow velocities the energy input into the plate results in an additional movement of the plate in the form of a secondary harmonic, with the largest effect shown for the case with the combined highest Reynolds-number and displacement, at $Re = 1512$ and $w = 0.002m$.

This strain, kinetic and total energies of the plate are shown in Figure 6.7. The results show a linear energy increase. The plate energy increases with the increase of the plate displacement and the flow speed. The figures show the strain energy, kinetic energy and total energy of the plate. The strain energy comes from the plate restoring forces, while the kinetic energy results

Figure 6.5: Wall displacement time series for the two inlet coupled cases

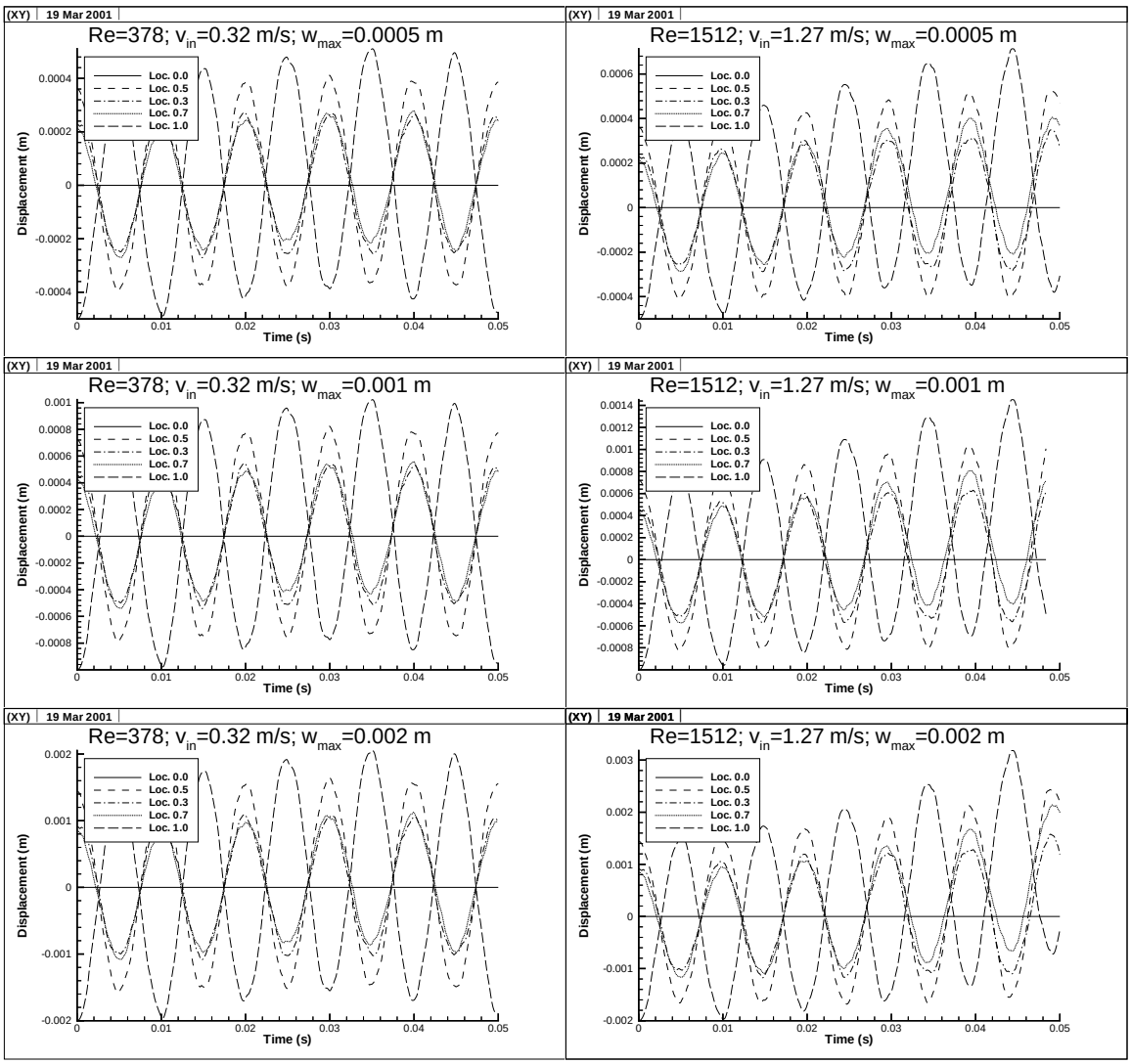
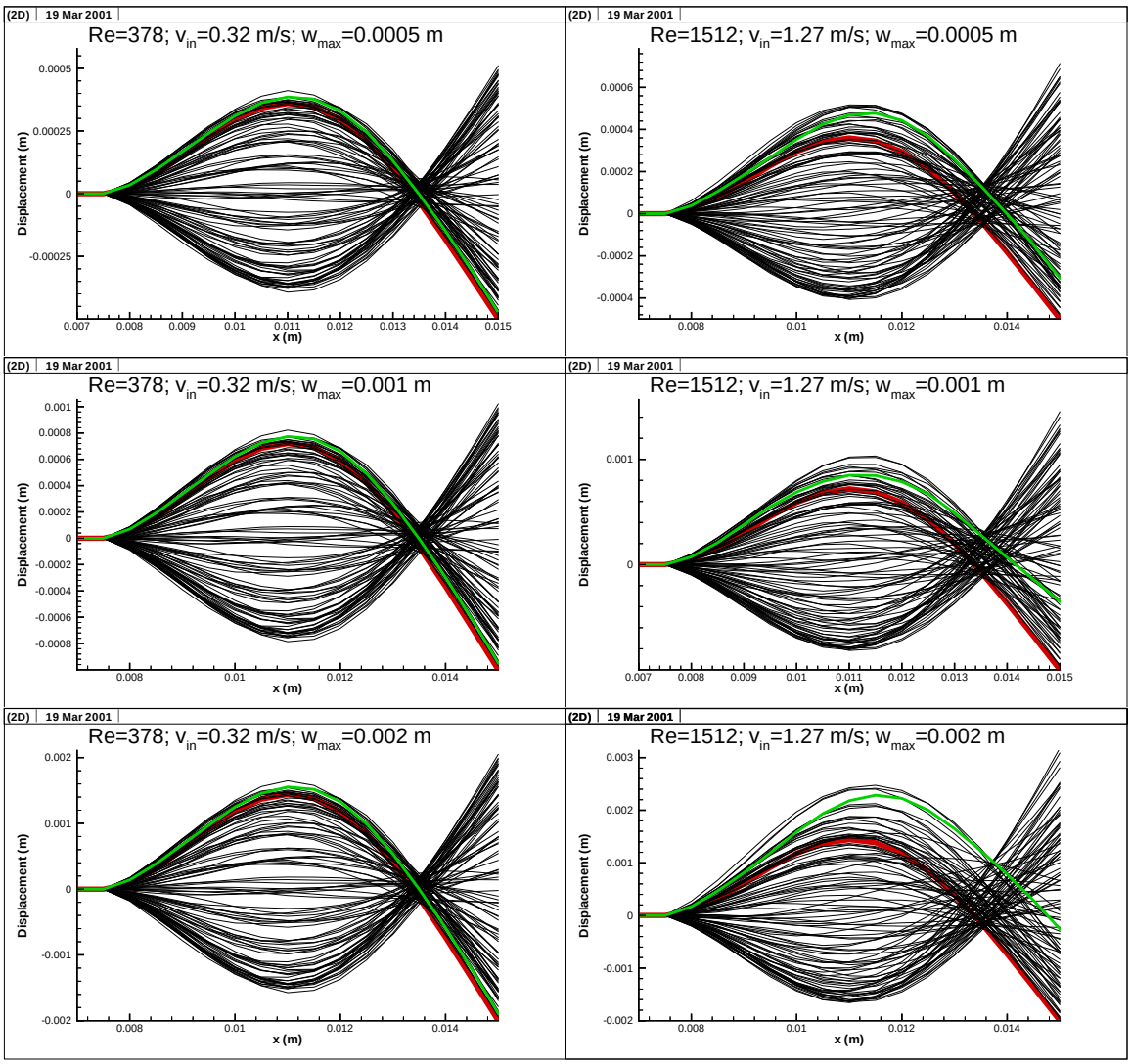


Figure 6.6: Plate movement time series for the two inlet coupled cases



from the plate movement.

As mentioned earlier, this energy input from the fluid to the plate results from a phase shift between the plate displacement and the pressure acting on the plate. Time series results for three cases are given in Figures 6.9, 6.10 and 6.11. The figures show displacement, velocity and transmural pressure through one oscillation cycle at four nodal locations. These locations are: Node 6, located between the fixed end and the local maximum in the middle of the plate; Node 9, is the local maximum in the middle of the plate; Node 12, located between Node 9 and the intersection of the x-axis at $y = 0$ and the plate; and Node 16, is the node before the free end of the cantilevered plate (see Figure 6.8). It can be seen that the phase shift is non-uniform along the flexible plate for all three cases. The phase shift changes between $30^o - 90^o$, $20^o - 70^o$, $0^o - 50^o$ and $110^o - 150^o$ for Nodes 6, 9, 12 and 16, respectively, based on inlet velocity and initial maximum plate displacement. (For comparison purposes the displacement, velocity and transmural pressure for all nine two inlet cases are shown in Figure 6.12, for Node 9 only. It can be seen that at $v_{in} = 0.32$ the pressure is about 70 degrees offset from the plate displacement. As the inlet velocity increases, this offset further reduces. At $v_{in} = 0.64$ and $v_{in} = 1.27$ the offset is about 45 degrees and 0 degrees, respectively. This trend is shown in Figure 6.13.) During flutter work is done on the plate, which can be calculated form Equation 3.84. Figure 6.14 shows that the work done on the plate increases with the increase of the Reynolds-number and maximum plate displacement. With the increase of the inlet velocity the work done on different sections of the plate also varies. To demonstrate this, the plate was divided into two sections. The upstream section was set from Node 1 to Node 9 (between the fixed end and the local maximum at the middle of the plate), while the downstream section was set from Node 10 to Node 17, (between the middle and the free and of the plate). As shown in Figure 6.15, with the increase of the Reynolds-number and also with increase of the maximum plate displacement, the work done on the upstream section of the plate increases faster than that of the downstram end of the plate. From the above argument, it can be concluded that the test

Figure 6.7: Energy time series for the two inlet coupled cases

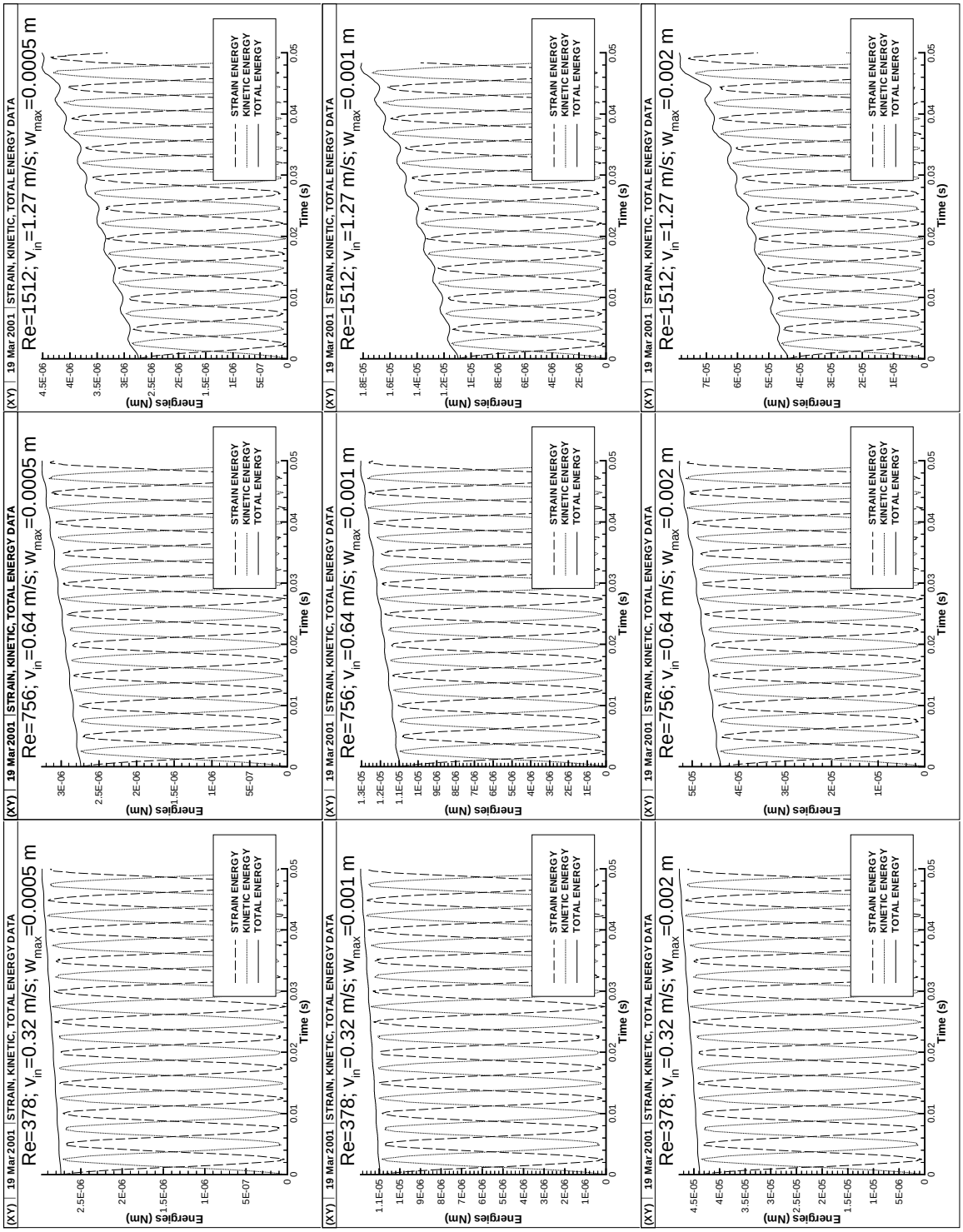
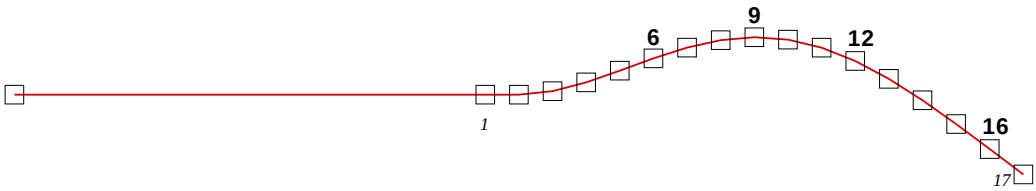


Figure 6.8: Flexible plate node numbering scheme



cases for a duct with two inlets successfully demonstrated the flutter condition through phase shift and work done on the plate.

For the one-inlet cases the wall time series are shown in Figure 6.16 for the 5 monitored locations. When the flow enters from one channel only, at the end of the cantilevered plate the flow velocity drops due to the sudden expansion. As explained with the Bernoulli effect, the flow velocity increases and the pressure drops in the upper open channel as the plate moves up and blocks the channel. At the same time the pressure in the lower blocked channel is higher than that for the two inlet cases. This higher transmural pressure between the two sides of the flexible plate forces the plate up as shown in the figures. When the inlet velocity is increased the plate becomes divergent to a point where the channel blockage exceeds the stopping criterion of the code. That is, when the blockage is larger than 80% of the channel height. Figure 6.17 demonstrates this through the plate movement cycle. Again, the red line indicates the initial displacement, while the green line shows the final position of the plate at the end of the time series. The cases on the right side (higher Re) clearly indicate divergence. As the plate displacement grows, its internal energy increases as shown in Figure 6.18.

As mentioned earlier, divergence occurs when the pressure forces on the plate outgrow the plate restoring forces. This is clearly visible in Figure 6.19. Note that in the figures the absolute values of the plate restoring and pressure forces were used, to precisely show when the

Figure 6.9: Case 1, Coupled time series for w, $\dot{w}$ and Δp, Nodes 6, 9, 12, 16

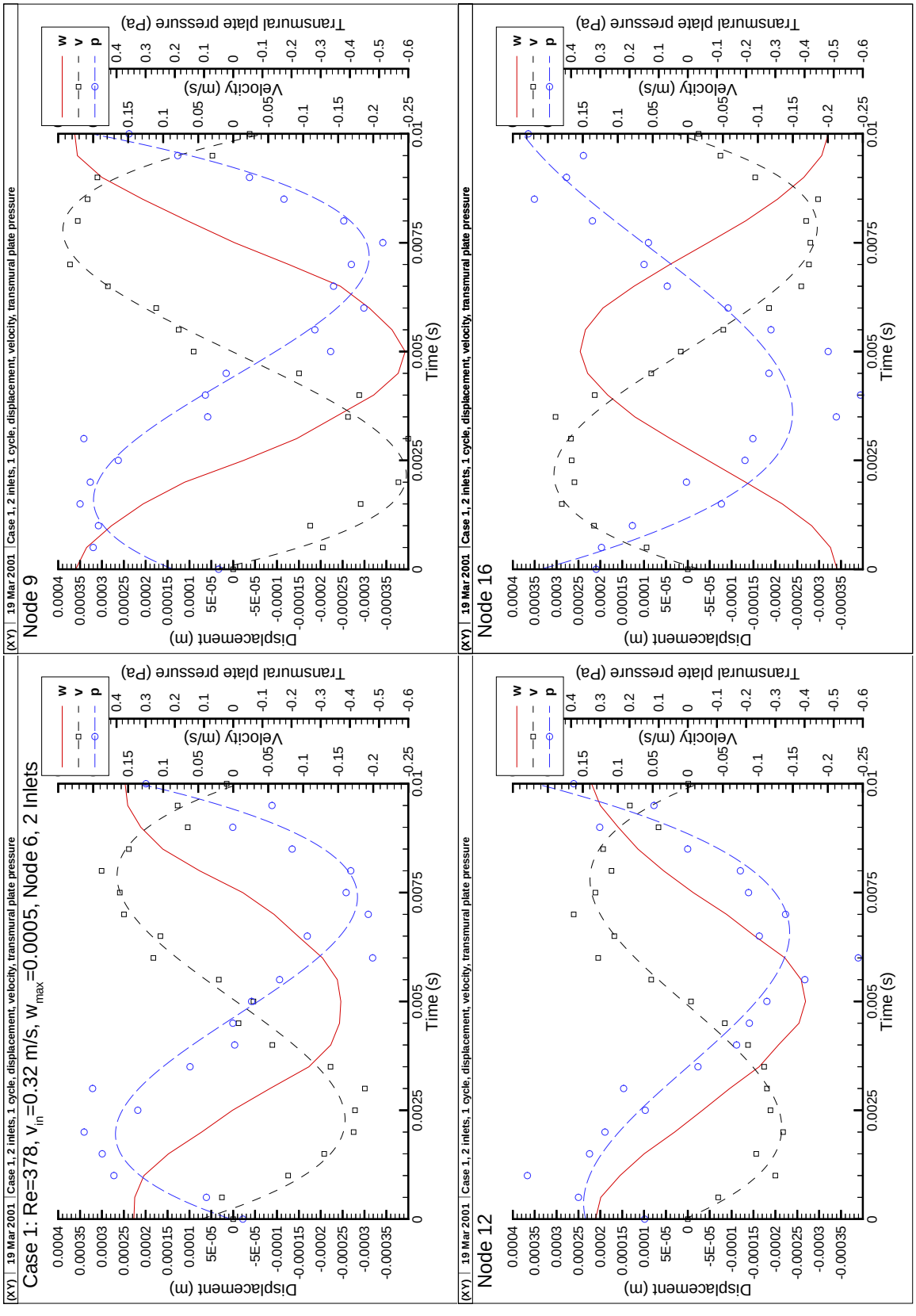


Figure 6.10: Case 3, Coupled time series for w, $\dot{w}$ and Δp, Nodes 6, 9, 12, 16

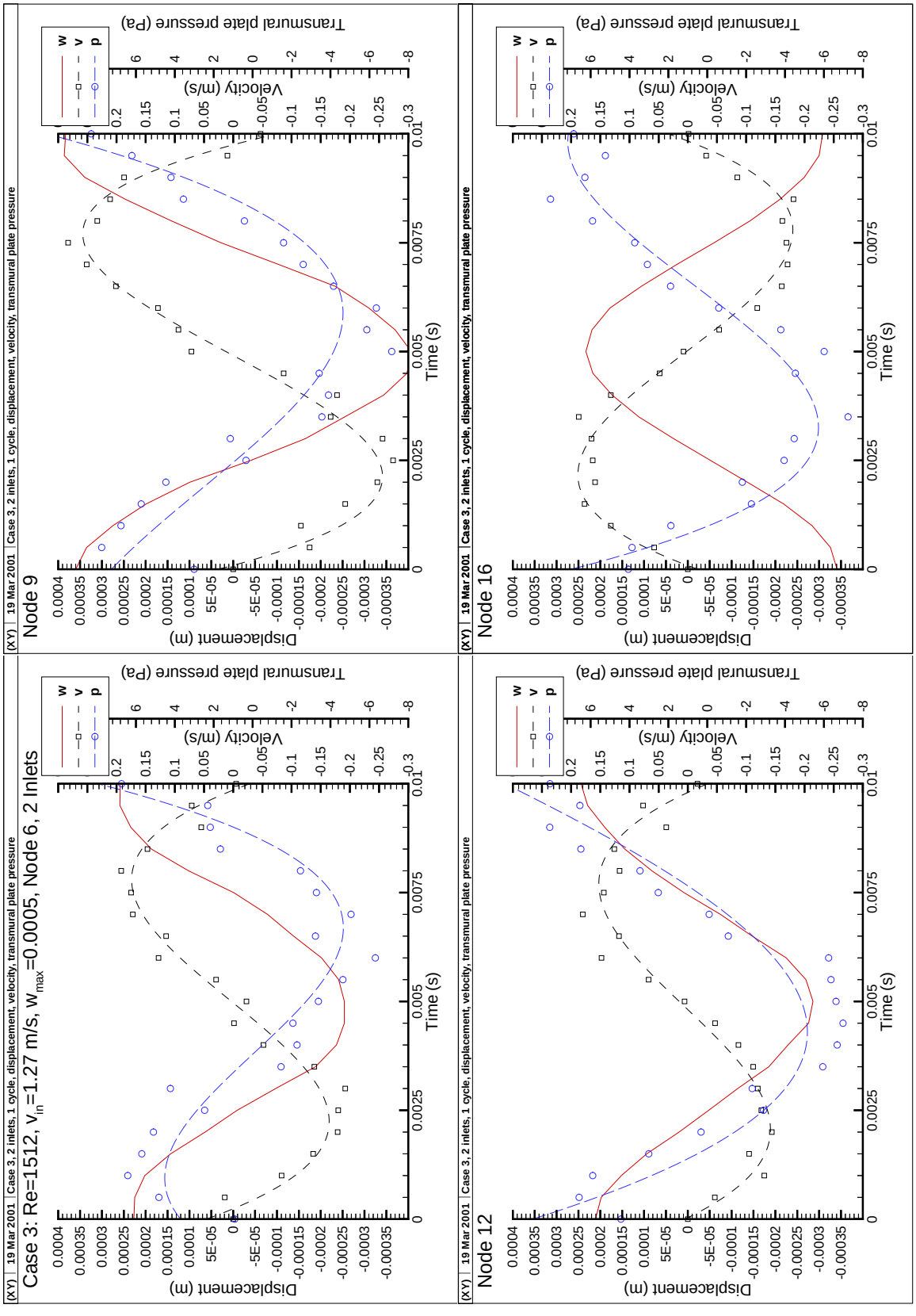


Figure 6.11: Case 13, Coupled time series for w, $\dot{w}$ and Δp, Nodes 6, 9, 12, 16

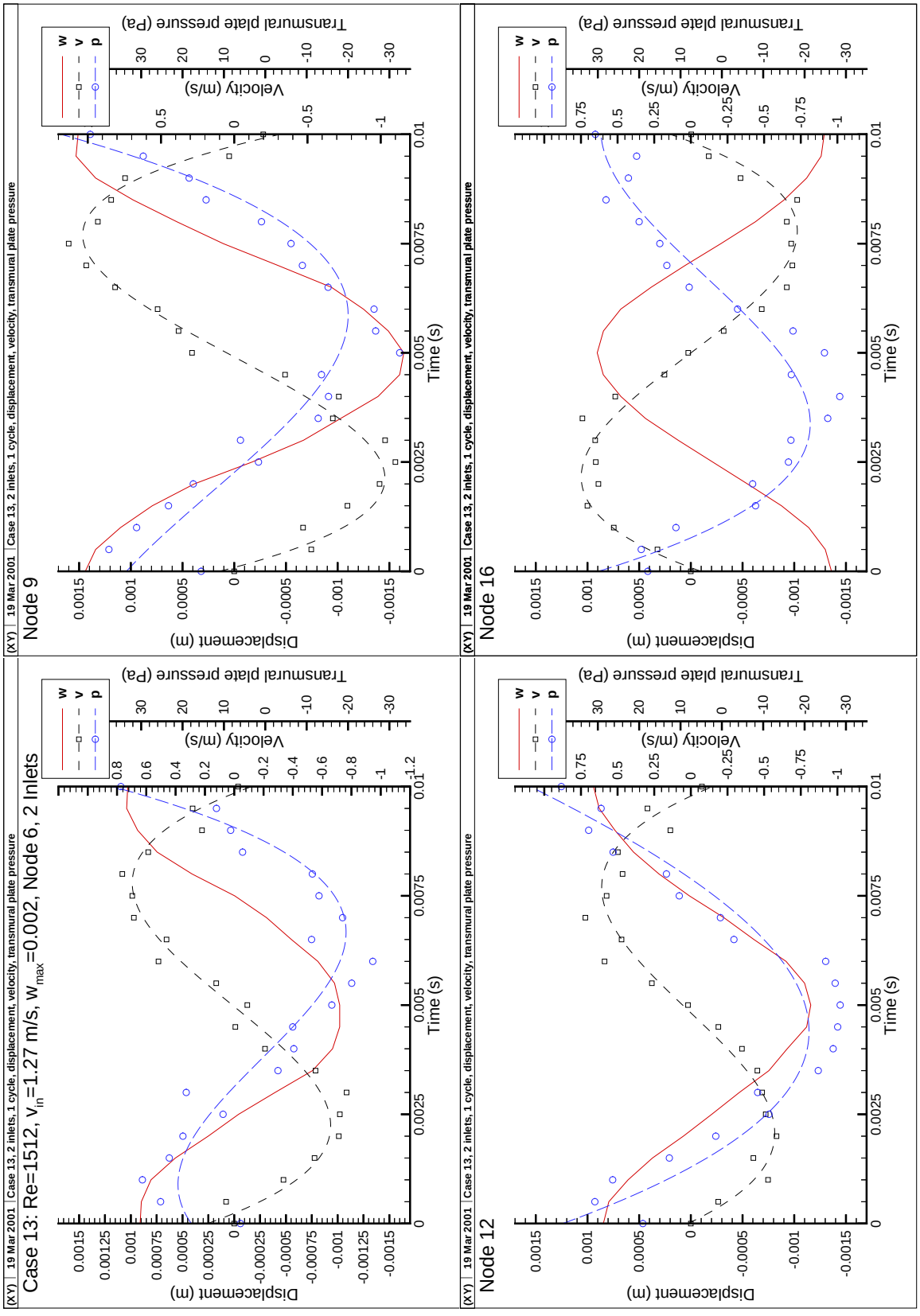


Figure 6.12: Coupled time series for w, $\dot{w}$ and Δp, Node 9, 2 inlets

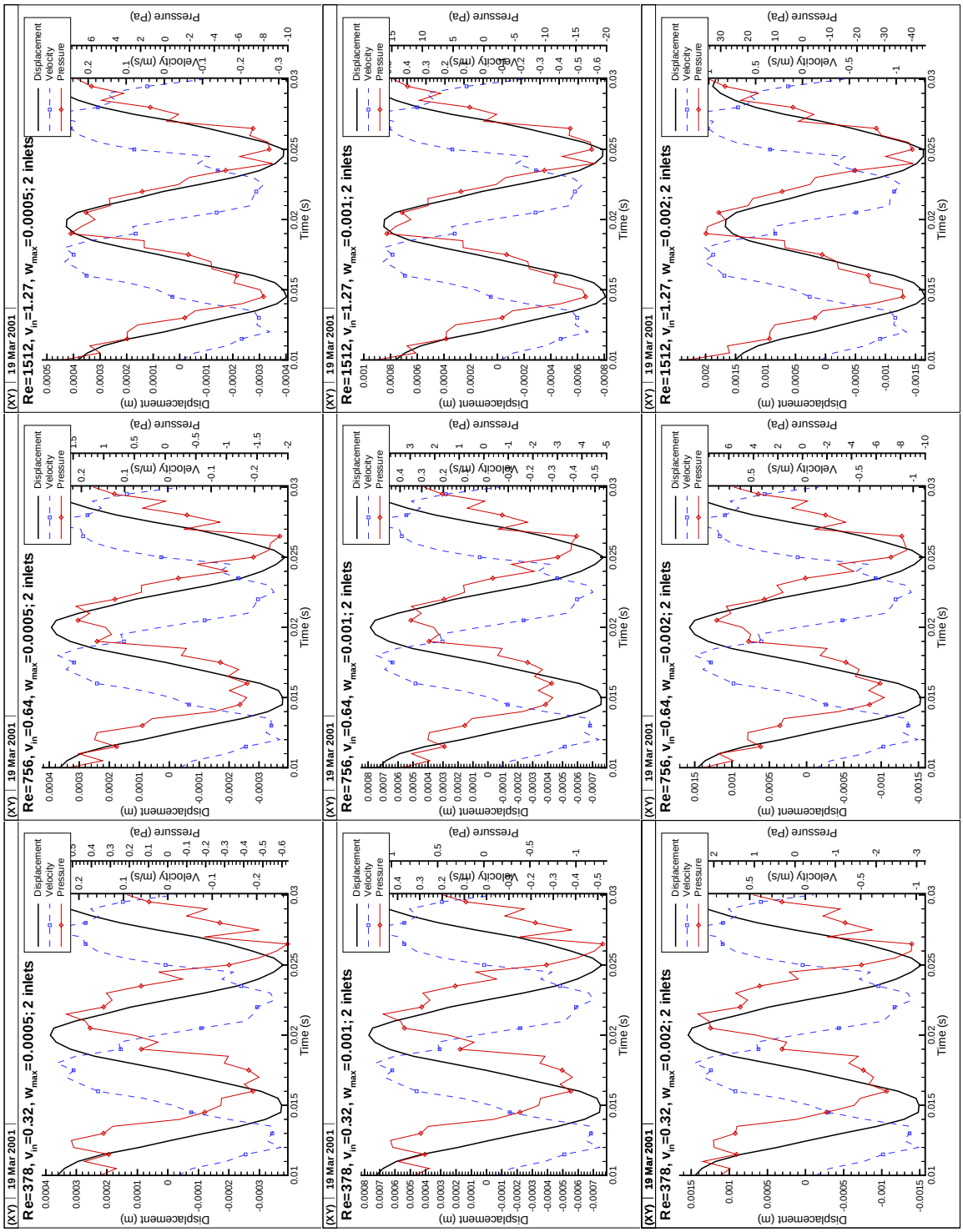


Figure 6.13: Transmural pressure phase shift at Node 9 as a function of v_{in}

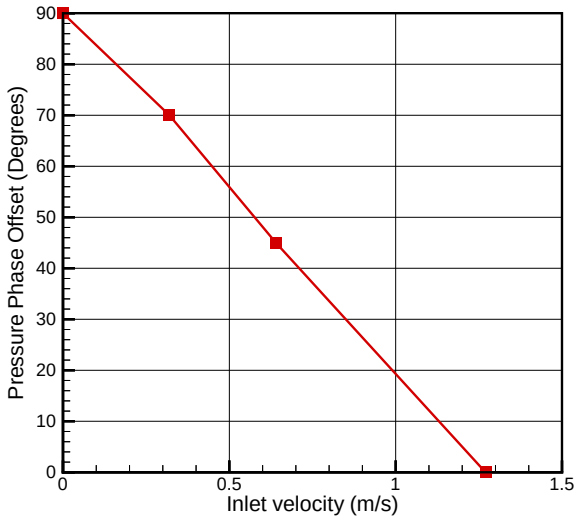


Figure 6.14: Work done on the plate

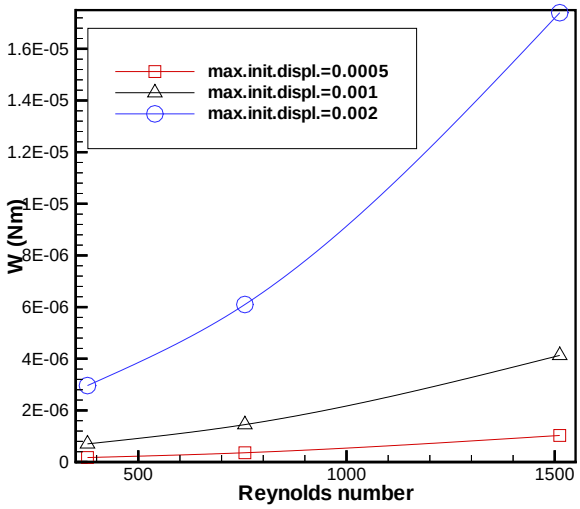
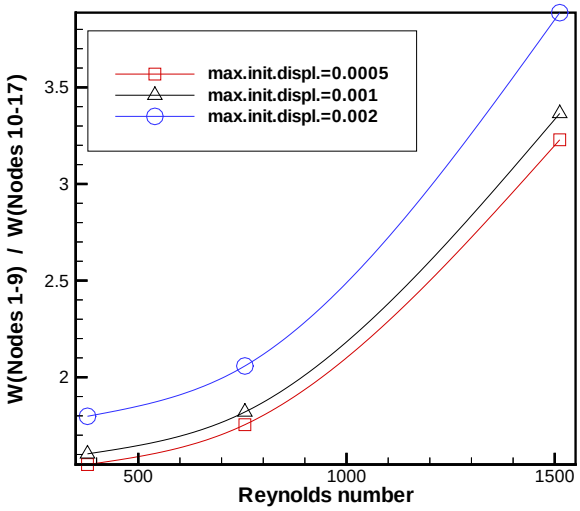


Figure 6.15: Work done on the plate - section ratios



pressure forces overtake the restoring forces. In the first case on the left hand side divergence is not present. With the increase of the maximum initial plate displacement, – see the 2nd and 3rd figures on the left hand side – the local velocities around the plate are increased due to continuity. For the 3rd case the development of divergence is obviously on its way. For all three cases on the right side divergence occurs as a result of increased pressure forces. (Note that on these plots the monitored node – Node 9 – is the one at the top maximum of the Mode 2 cantilevered plate.)

Finally, the total energies were normalized for all of the cases (see Equation 3.87). In Figure 6.20 the upper figure shows the energies for the two inlet cases, while the bottom figure for the one inlet cases. For the two inlet cases it was found that the normalized total energy transients collapse into the same lines at the same inlet velocities and Reynolds-numbers. For the three sets of inlet velocities the total energy doubles in the first instance, but when the inlet velocity is doubled again, the total energy quadruples and even starts to exhibit non-linearity in its

Figure 6.16: Wall displacement time series for the one inlet coupled cases

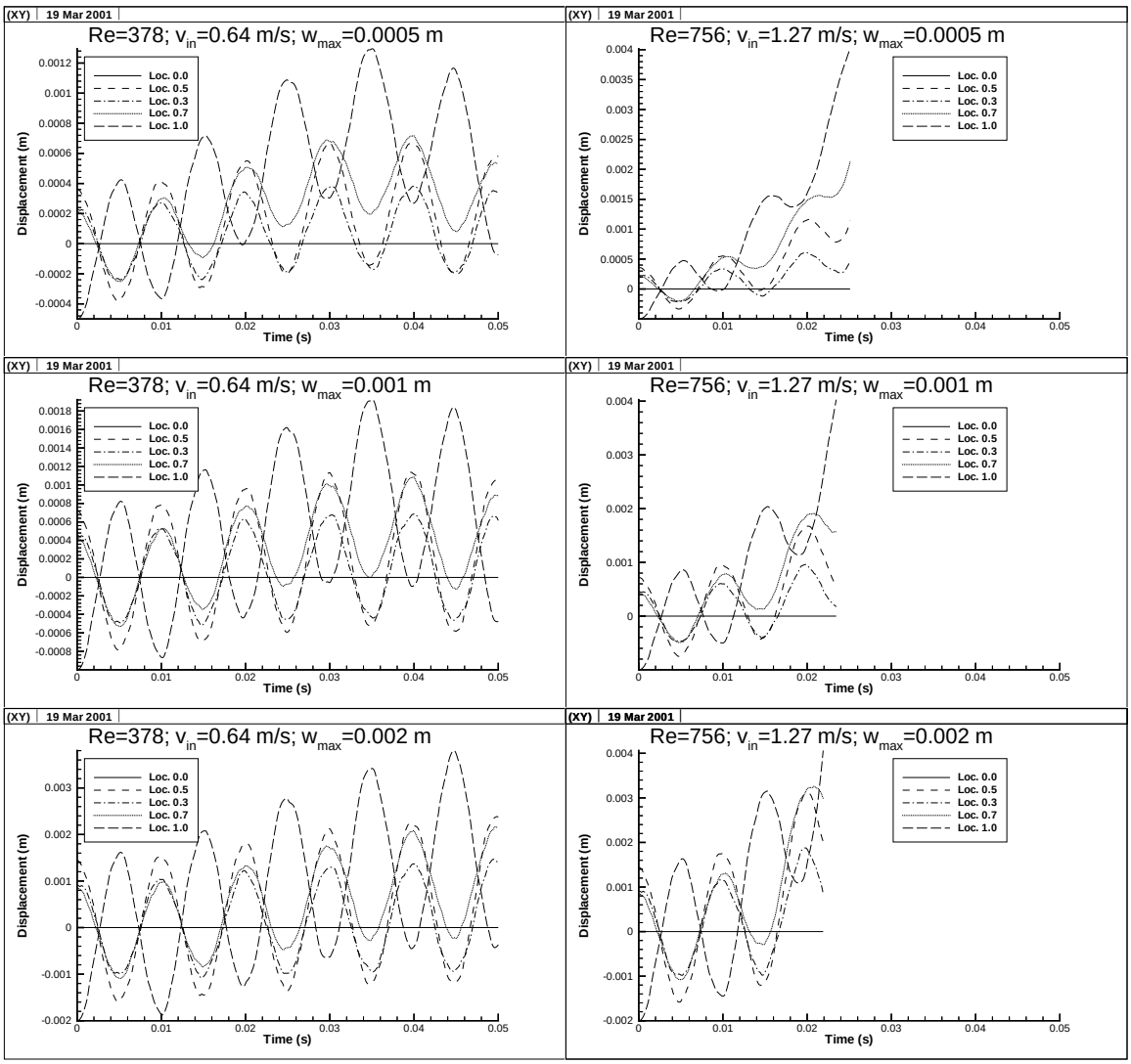


Figure 6.17: Plate movement time series for the one inlet coupled cases

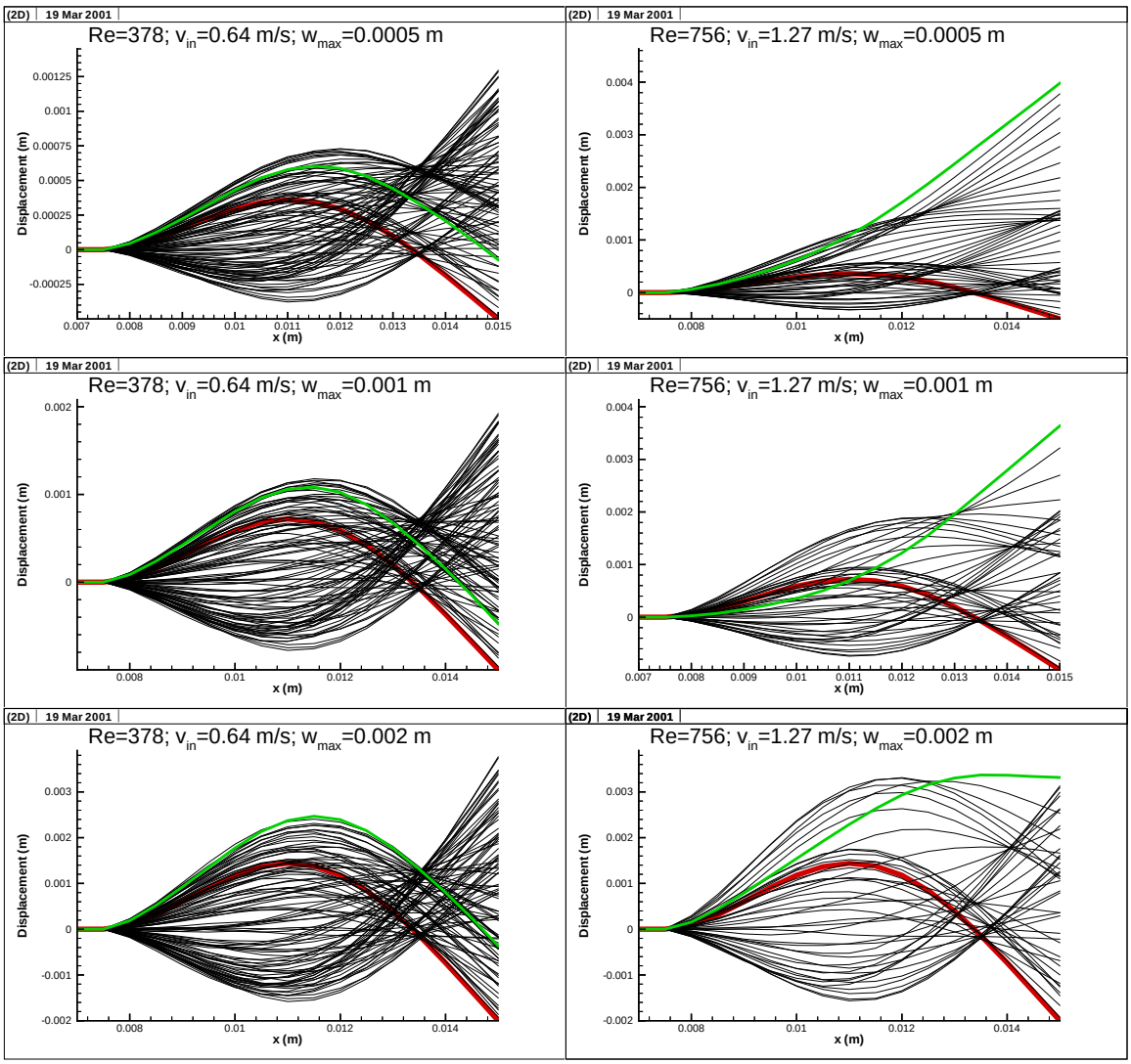


Figure 6.18: Energy time series for the one inlet coupled cases

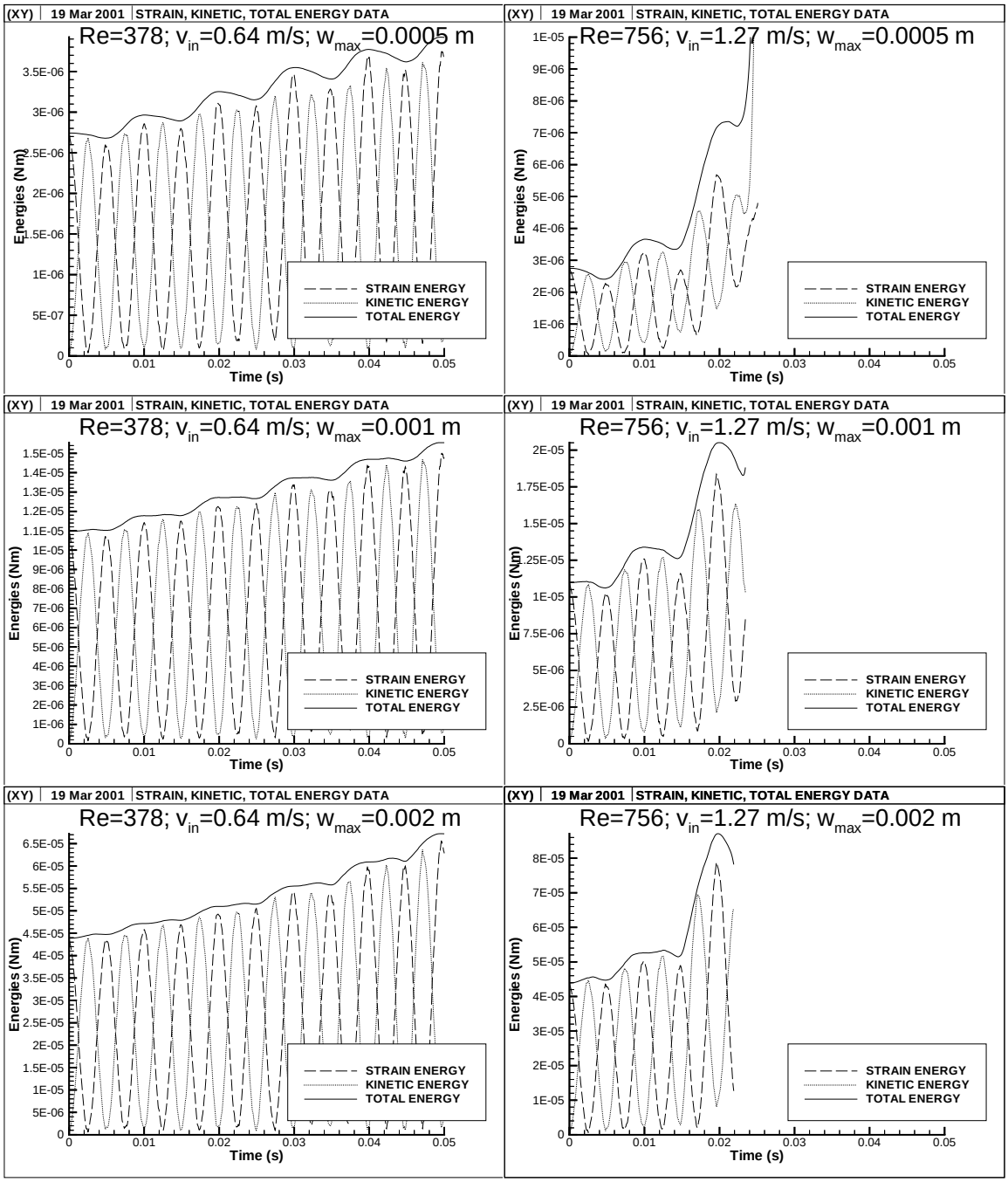
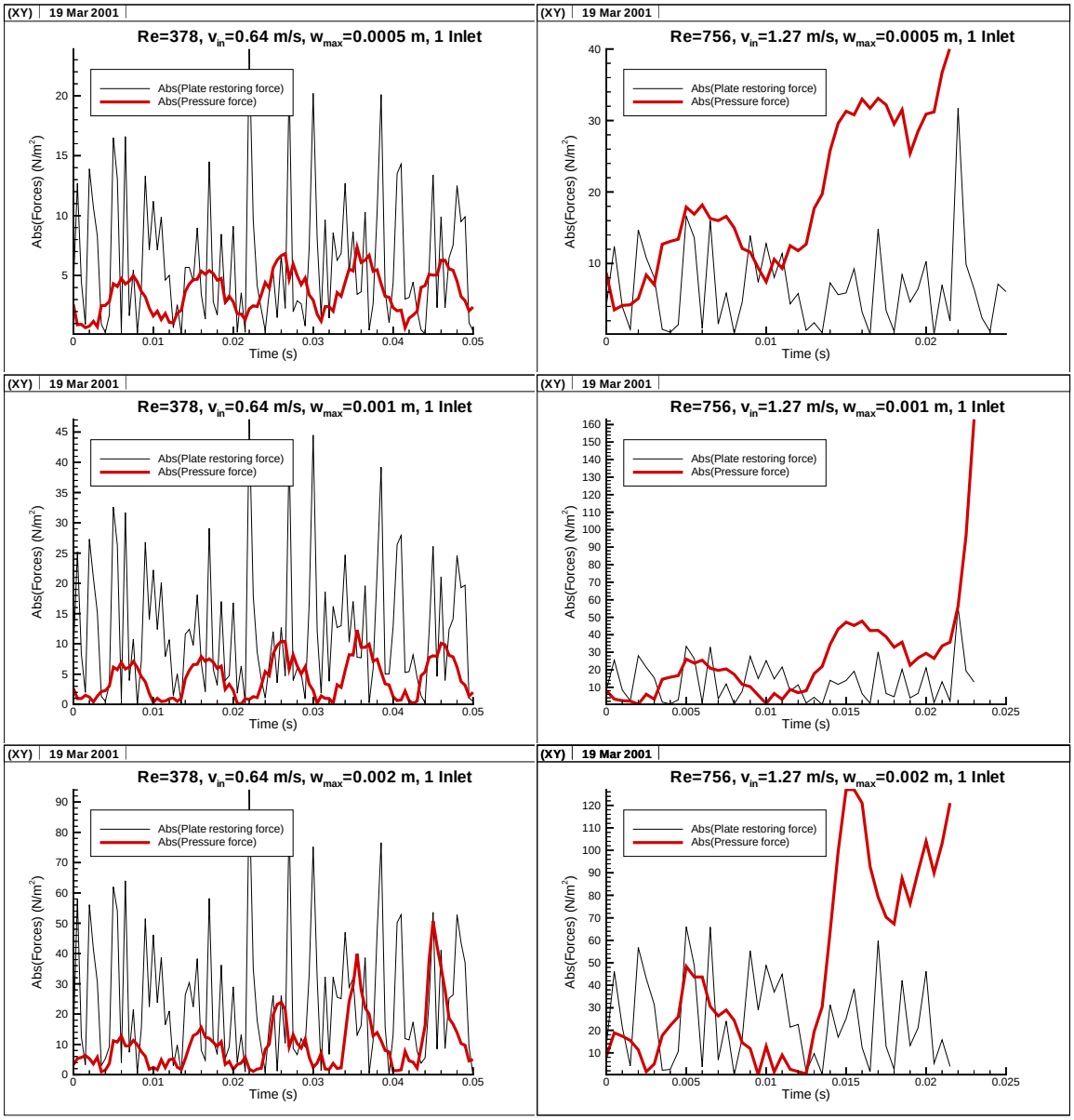


Figure 6.19: Plate restoring and pressure force time series, 1 inlet (absolute values)



growth rate towards the end of the time series. For the one inlet cases it is obvious that at higher Reynolds-numbers the energies grow out of bound due to divergence. Therefore, it can be concluded that for the one inlet cases a divergence conditions is successfully demonstrated.

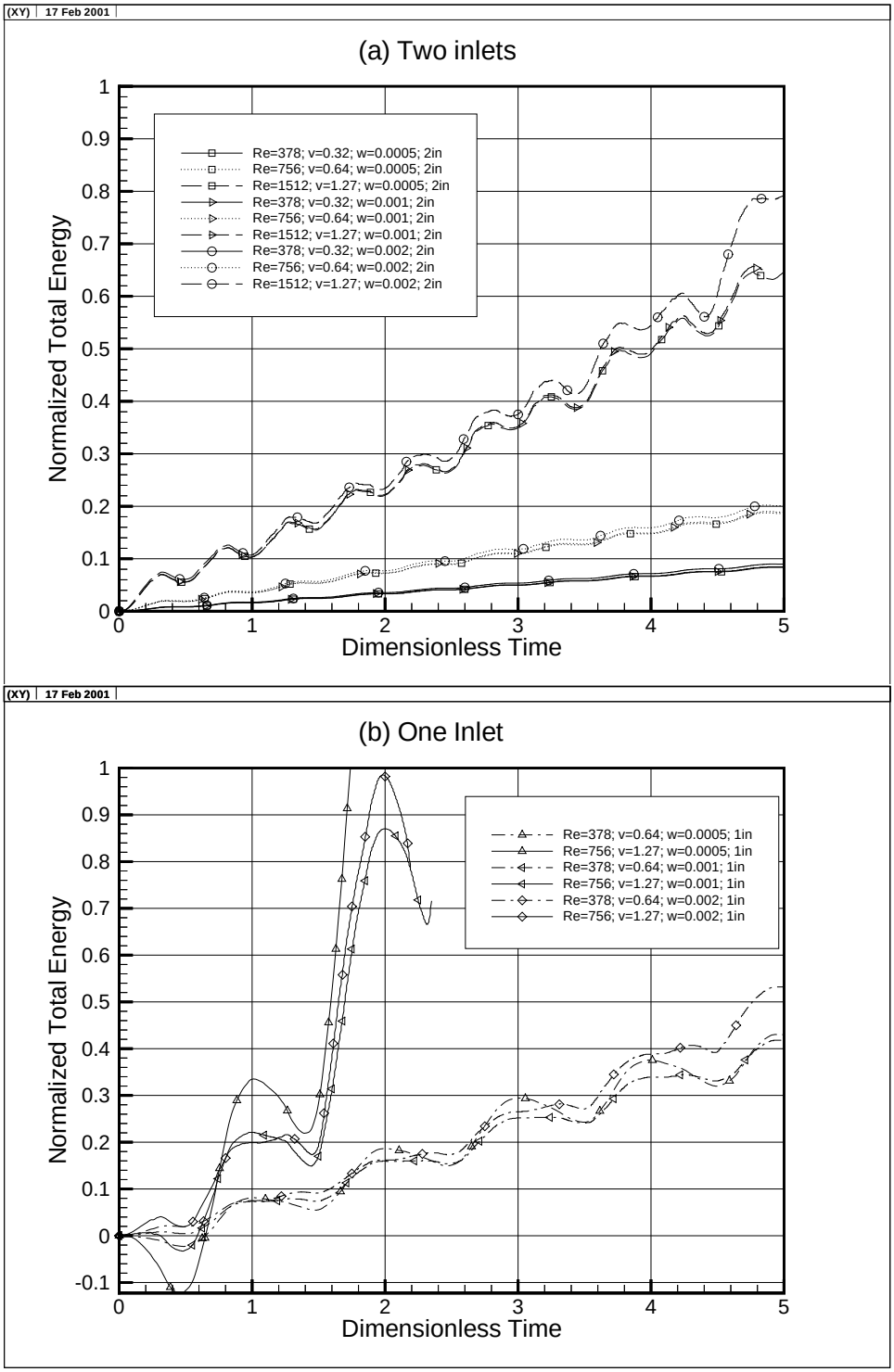
6.2 Coupled vocal tract/soft palate model

An additional run was performed to demonstrate the applicability of the code on more complex geometries, representing the vocal tract with a compliant cantilevered section.

The mesh generator was modified to account for the varying local duct cross sections and for the moving soft palate. Once the mesh generator was created, the code coupling worked the same way as explained in the previous section.

The physical boundaries of the vocal tract were the same as those used for the vocal tract model with rigid walls. The length of the duct was set to $L = 0.17458m$. The domain had 80 elements along the x-axis and 16 elements along the y-axis. The length of the combined soft and hard palates were 37 elements, from which the soft palate accounted for 18 elements. The palate split the channel in the middle, 8 elements from both the top and the bottom. The fluid density was $\rho_f = 1.1774kg/m^3$. The Reynolds number was $Re = 756$, with a uniform inlet velocity to both channels at $v_{in} = 0.64m/s$. One cycle of oscillation was calculated. Just as before, the plate frequency was set to $100Hz$. However, due to the longer plate length – compared to the straight duct cases – the wall-code stability required a reduced time step, which was also used for the fluid code ($t_{end} = 0.008s$, $\Delta t = 4 \times 10^{-5}s$). The plate properties were also reset to the following: thickness $h = 0.00001m$, wall density $\rho_w = 2477kg/m^3$, elastic modulus $E = 6.347 \times 10^{+11}N/m^2$, Poisson ratio $\nu = 0.333$, Mode 2 initial plate displacement, with $w_{max} = 0.001m$. The stopping criterion was set to $RMS_W = 1 \times 10^{-15}$.

Figure 6.20: Normalized total energies time series for the coupled cases



An example of the time series is given in Figure 6.21 for the first cycle of oscillation – in 9 steps. (The figure shows the u_1 velocity components.) At the end of the time series the displacement reached the stopping criterion, i.e., the flow blockage exceeded 80% of the duct height. The flow enters from the left side in two channels with a uniform inlet velocity. As the vocal tract expands, the flow slows down, then it accelerates around the soft palate as the channel narrows. When the end of the palate moves upward, the flow velocity increases in the upper channel. After reaching a maximum displacement, the palate moves back. During this time the flow accelerates in the other channel. Downstream from the palate the channel further narrows, and due to the small diameter the flow accelerates to a point, where the upstream conditions do not seem to have an effect on the local flow behaviour. A summary of the time series results, such as displacement and energy plots are shown in Figure 6.22.

Note that the discretization for this geometry is very coarse. This resulted in elements skewed beyond reasonable levels and element sizes which were too big to capture the boundary layer, further eroding the accuracy of the predictions. Nevertheless, the methodology introduced here is suitable for such calculations. It is expected that when the grid size is sufficiently increased, such a setup will yield the desired prediction accuracy.

6.3 Conceptual design for pressure driven flows

The present numerical experiments were performed using a uniform inlet velocity boundary condition. A code using a velocity-pressure numerical formulation gives the velocity field for each time step, and as a by-product calculates a corresponding pressure field. For a given cantilevered plate displacement, this results in a changing channel resistance and consequently in a changing pressure drop across the duct. This use of the inlet velocity as a boundary condition is a simplification, since in reality the flow through the upper airway is pressure driven. During air intake, the lungs generate a negative pressure and consequently a pressure drop is

Figure 6.21: Coupled time series cycle, Vocal Tract, $Re = 756$, $v_{in} = 0.64m/s$, $w_{max} = 0.001m$

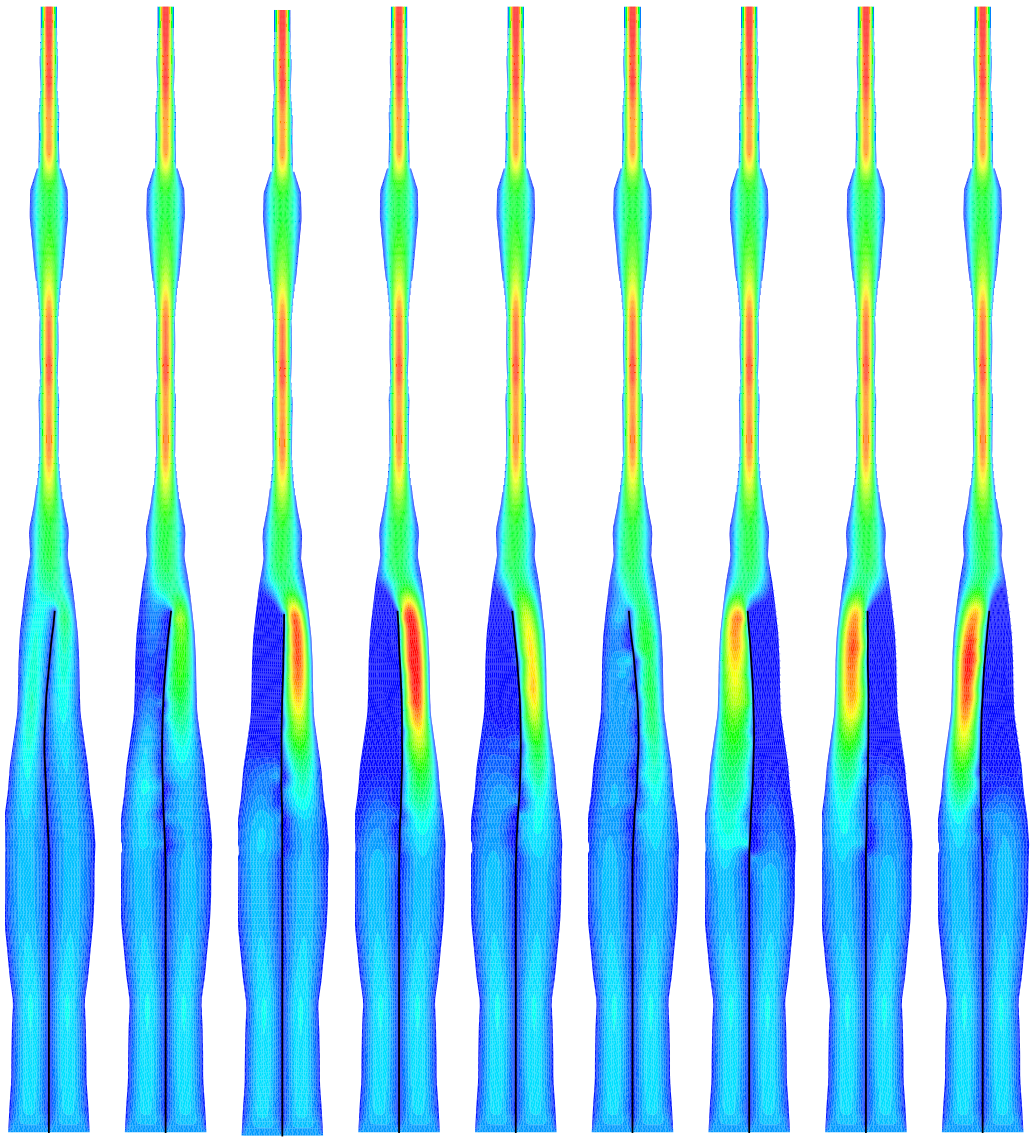
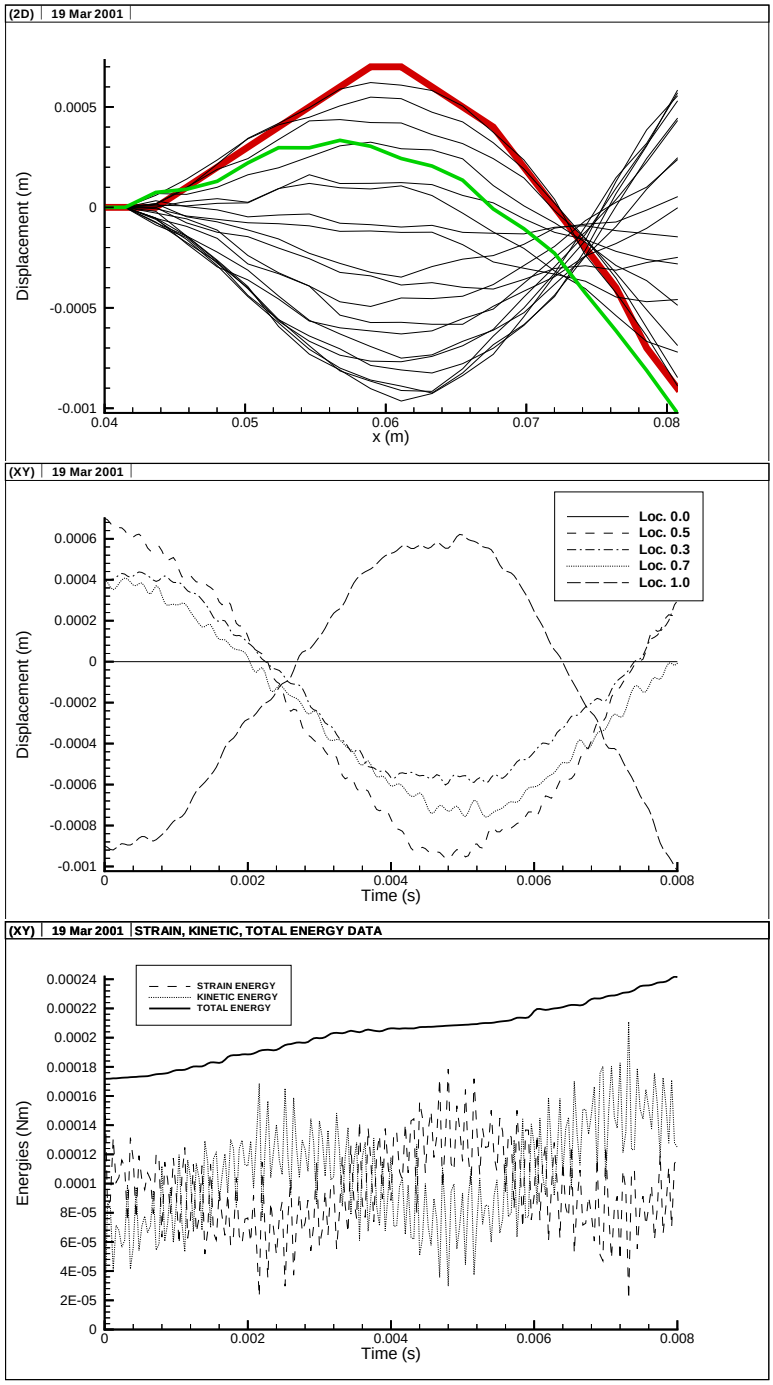


Figure 6.22: Coupled time series summary, Vocal Tract, $Re = 756$, $v_{in} = 0.64m/s$, $w_{max} = 0.001m$



achieved between the lungs and the atmospheric pressure outside. (A summary of the lung volumes, capacities and pressures are given in Appendix B.) This pressure drop through the upper airway is the driving force for the air, where the air-flow velocity is based on this pressure drop and the airway resistance. The pressure drop is non-uniform, it changes throughout the breathing cycle with the changing air intake.

As an initial modeling approximation it can be assumed that the pressure drop is uniform during the time series. Also assuming single inlet and outlet channels – with one of the inlet channels blocked – an iteration loop can be implemented in the fluid code. The iterations are used to achieve the required pressure drop between the channel inlet and outlet, based on the channel resistance. The resistance is directly related to the cantilevered plate displacement and consequently to the channel resistance. On a cross-sectional average basis the channel pressure drop can be calculated from

$$\begin{aligned} \Delta p &= \left(f\frac{L}{D_h} + \sum K \right) \frac{G^2}{2\rho_f} = K' \frac{G^2}{2\rho_f} \\ &= \sum K \frac{\rho_f U^2}{2} \qquad (f = 0) \end{aligned} \tag{6.1}$$

where f is the skin friction factor, L is the duct length, D_h is the hydraulic diameter, $\sum K$ is the sum of the form losses, K' is the overall loss coefficient, $G = \rho_f U$ is the mass flux through the duct, U is the cross-sectional average fluid velocity and ρ_f is the fluid density. The present fluid model does not account for the skin friction in the channel, therefore, the first component of the equation is neglected ($f = 0$). The form losses are based on the channel obstruction by the cantilevered plate. It can be seen that with the increase of the plate displacement the loss coefficient increases in the blocked channel. To maintain a constant pressure drop, the the inlet velocity must be reduced. The inner iteration loop inside the fluid code should be maintained until the required constant pressure drop is reached.

For a configuration with two inlet channels the situation is more complex. For a given dis-

placement of the cantilevered plate the flow resistance changes in both channels. While one inlet channel with a reduced cross sectional flow area increases its resistance, the other channel opens up and decreases its resistance. The pressure drop across the duct inlet and outlet remains constant, while the inlet velocities change, proportionally to the channel resistances. It should be noted that these inlet velocity changes can influence the plate stability. The reduced channel flow results in a larger local pressure, while the increased flow in the other channel reduces the local pressure. Therefore, the pressure acting on the plate has a reduced destabilizing effect from that of the plate pressure based on constant inlet velocities.

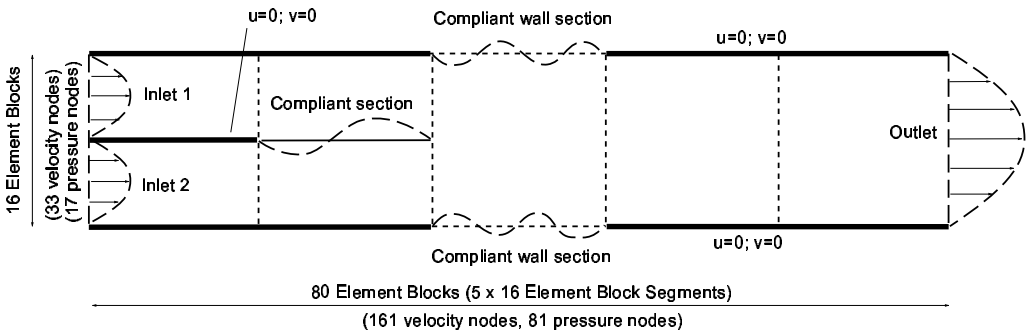
In future studies this phenomenon needs to be addressed.

6.4 Conceptual design for domains with multiple compliant sections

A model of the upper airway with multiple compliant sections is shown in Figure 6.23. The flow enters in two channels, an upper – representing the nasal inlet – and a lower – representing the oral entrance. In the first segment the two inlet channels are divided by a rigid wall representing the hard palate. The next compliant segment is analogous to the soft palate. The third segment contains the compliant wall sections of the pharynx. The last two rigid undivided segments before the exit help with the downstream flow development.

The coupled code would be similar to the one outlined in the previous subsections, however, the multiple compliant sections represent a complication to the method. While the soft palate frequency is between 30 and $100Hz$, the frequency of the pharyngeal region is around $2-3Hz$. Therefore, in the modeling the wall stiffness should be reduced to achieve this frequency. When the stiffness is reduced, the code stability increases, i.e., the time step could be increased. This is an advantage, since both the fluid and the wall codes use the same time step. Therefore, the same time steps can be used for the compliant pharyngeal region as the one used for the can-

Figure 6.23: Split channel with compliant sections



tilevered plate and fluid code segments, without jeopardizing numerical stability or extending the execution time due to the smaller time steps. The algorithm for the coupled code is similar to the one with the cantilevered plate only, but the code must call up the wall calculations three times, once for the cantilevered plate and twice for the pharyngeal region, for the top and bottom wall sections, respectively. The wall pressure is based on the pressure difference between the top and bottom sides of the plate. For the cantilevered plate the transmural pressure is the nodal pressure difference between the corresponding nodes in the top and the bottom inlet channels. For the pharyngeal region, one side of the plate is inside of the channel, while the other is on the outside. Another issue is how to satisfy the continuity equation. When the compliant cantilevered plate is moving towards the lower or upper channel walls, the downstream pressure loading on the pharyngeal plates is uneven. This results in a plate movement, which is different for the two plate sections. Off phase movement of the plates may create a changing domain volume between time steps. This does not satisfy the continuity equation. If the movements of the plates are set to be cyclical, it can be assumed that the time average of the volume change is constant. While it results in an error between time steps, for long time series the continuity equation is satisfied. The error can be reduced by using very small time steps, where the change in domain volume between time steps is negligible. Further information on this problem can be found in the works of Luo & Pedley [LP98] [LP96] [LP95].

Chapter 7

Summary and Recommendations

7.1 Summary

The thesis documented here described a systematic examination of the flow–structure interaction in a two-dimensional duct, representing the upper airway. The study was divided into three major sections. These were; (1) the duct flow problems with rigid walls, (2) the linear elastic wall deformation problems and (3) the coupled flow–structure interaction problems.

7.1.1 Navier–Stokes problems

The duct flow calculations were based on the Navier–Stokes equations. The equations are solved using the Finite Element Method. Code validation was performed through test cases for various benchmark geometries, such as for lid-driven cavity flows (in triangular of rectangular domains), and for two-dimensional duct flows, such as for Poiseuille flow, for backward and forward facing steps, for a vertical fence in the duct, split inlet domains with two open inlets or with one inlet blocked. The domain discretization was performed using fast mesh generators for each tested geometry. These mesh generators were specifically written for this research. Spatial and temporal convergence of the method was examined for the lid-driven cavity flow cases. In addition, comparisons were made to published benchmark results for the backward facing step cases. It was found that the code accurately predicted the flow conditions at low Reynolds-numbers. While the flow phenomena was captured in the solutions, such as the re-circulation zones and pressure distribution due to geometric variations, the accuracy was

noticeably reduced with the increase of the Reynolds-number. The main reasons for this were two fold. First, the FEM method used did not include a turbulence model. Second, the grid size was limited by the physical memory of the computers, used for the calculations. The mesh size was maximized by the development of a systematic array allocation and deallocation, a programming feature introduced in FORTRAN 90. However, this still coarse grid did not always capture the boundary layer along the walls. While at low Reynolds-numbers the grid resolution was enough to capture the boundary layer, achieving good agreement with open literature data, at high Reynolds-numbers the boundary-layer was significantly thinner than the element height. Beside the obvious disadvantages, the approach provided several advantages. The limited grid size allowed for the calculations to be performed on desktop computers. The explicit model provided numerical stability. The smaller mesh size also reduced the code execution time. Shortened axial flow structures – such as re-circulation zones – resulted in a flow time series, which contained the flow phenomena within the short domain length. Note that the results did not capture vortex shedding, again due to the coarse grid. It can be argued that for the present configuration and the phenomena studied the upstream effect of the vortices can be neglected. In light of these findings, it was found that the numerical model used here was suitable to investigate fluid–structure interaction problems.

7.1.2 Flexible plate problems

The elastic deformation of the plate was calculated using the finite difference method. The code was validated against analytical solutions for two configurations. In the first instance both ends of the plate were fixed, while in the second set of numerical experiments a cantilevered configuration was employed. The end conditions were hinged, clamped or free. The analytical and numerical results in vacuo proved to be in excellent agreement. In addition to in vacuo cases, the use of damping and/or pressure forces were also enabled. The sensitivity of the results due to discretization were examined and the results were shown in phase diagram plots. As expected, the accuracy improved at higher discretizations. It was found that the code accurately predicted the nodal wall displacement and velocity during the time series and it

responded to pressure forces, making it a good candidate for code coupling.

7.1.3 Fluid–structure interaction, using a fully coupled code

Preliminary investigations of eigenmode displaced plates in a duct, under quasi-steady-state conditions, demonstrated a non-uniform pressure drop distribution along the plates. Full coupling between an explicit FEM code for for fluids, solving the Navier-Stokes equations, and an implicit finite difference code for the compliant wall, solving the equations for linear elastic deformation were achieved, using a method developed for this research. For the fully coupled cases with two inlets the results showed a phase shift between the displacement and pressure transients, indicating a flutter condition. For cases with one inlet blocked, the pressure forces gradually outgrew the wall restoring forces, pointing to a divergence condition. A sample case was also calculated for a more generic physical geometry, in the form of a simplified two dimensional model of the upper airway. Based on wall energy transients, wave growth rates were calculated for the plate flutter under various initial and boundary conditions. With the increase of the Reynolds-number the wall displacement growth rate started to exhibit an exponential trend. However, it was found that further improvements to the numerical scheme are required to refine the results, to allow for higher discretization, higher Reynolds-numbers and more complex geometries.

These investigations provided encouraging results regarding the flow–structure interaction in the upper airway. Such a study extends our understanding of mechanisms in the upper airway under normal and abnormal physiological conditions. These include snoring and Obstructive Sleep Apnoea / Hypopnoea. During snoring, flutter of the soft palate translates to audible sound. Flow induced instability due to divergence obstructs the upper airway, resulting in its partial or full blockage. Although the present research demonstrated these mechanisms, a more extensive study is required to understand fully the behaviour of the upper airway under these conditions.

Based on the current findings, recommendations for future work are given in the following subsection.

7.2 Recommendations for future work

7.2.1 Code development

Mesh generator

The mesh generator is a very important part of the code. The present model used a simplified straight two dimensional model of the upper airway. This model can be further improved to represent the upper airway more closely. Some of the recommended improvements are listed below:

- Following up on the present research, additional compliant sections could be incorporated into the model to represent the pharyngeal region of the upper airway. This would further complicate the re-meshing process, but would result in a more complete model of the upper airway.
- To make it more realistic, curvature could be introduced to the upper airway model, where the streamwise inlet and outlet velocity vectors are offset.
- An un-structured grid would allow for more geometric flexibility and local grid refinements.
- A three dimensional grid could further improve the realism of the model. However, it would significantly increase the mesh size and consequently the memory requirement and code execution time.

Fluid code

Improvements to the fluid code can be made by improving the numerical model or by improving the physical model. Some of these recommendations are listed below:

- Migrating the code to a UNIX machine would improve the computations resources. However, this would require a complete re-write of the code to match the input requirements of the new mathematical library functions, for example that of a new sparse matrix solver.

- In the present model, the elemental and global matrices are assembled first, then the global matrix is sparsed. This requires a peak memory usage before the global system matrix is deallocated from memory. The memory requirement of the numerical model can be reduced – and therefore the discretization improved – by developing a new algorithm, where the system matrix is sparsed directly from the elemental and global matrices.

- Alternatively, the direct method can be replaced with a conjugate gradient type method (for example Uzawa's conjugate gradient method) [RW94]. This is a matrix free implementation with small memory requirements. However, it uses inner iteration loops to reach convergence for the solution, resulting in complications especially when combined with an implicit method for the fluid code.

- Due to the transitional Reynolds-number range during the peak air intake, a turbulence model would be needed to properly model the phenomena.

Wall code

The code for calculating the elastic deformation of the plate can be improved by:

- A non-linear model would provide a more realistic representation of the soft palate, allowing for streamwise node displacements as well. This change should be made in conjunction with a suitable mesh generator.

Coupled code

Once the recommendations for the fluid and wall codes are implemented, the code coupling would require certain improvements as well. For example:

- The coupling algorithm should be re-written to reflect the pressure driven flow nature of the problem. This would require an inner iteration loop for the fluid solver, where the inlet velocity is varied until the pressure drop across the duct would converge to a constant value.

- The coupling algorithm could be re-designed for an implicit-fluid/implicit-wall model, in which case additional iteration steps would be needed in the solutions process.

- The soft palate and pharyngeal regions flutter at different frequencies. This should be accounted for using the wall properties for the various wall sections.

7.2.2 Future research direction

In the present analysis a preliminary study was performed to examine the fluid–structure interaction in the upper airway. It is recommended to extend the research, to gain better understanding of the phenomena by the following:

- The time series time could be increased to cover a larger number of oscillations. This could give better indications of periodicity, flutter and divergence behaviours.

- The breathing cycle in non-uniform. This could be modeled using unsteady inlet flow conditions. Combined with a longer time series, breathing and snoring cycles could be modeled using the method developed in this thesis.

- The test matrix should be extended to cover a wider range of physiological conditions. A breathing cycle requires Reynolds-numbers up to about $Re = 3500$. This is a transitional range, which may require the inclusion of a turbulence model.

- Further improvements to the mesh generator would allow for more complex geometries. These could include multiple compliant sections, blockages and a curved upper airway to simulate the physical geometries more closely.

- The soft palate model could be improved. In this thesis a linear elastic deformation model was used together with thin plate mechanics. The thin plate model could be replaced with a two-dimensional plate, combined with a non-linear elastic deformation model.

- Finally, the same numerical method could be used to study blood flow in arteries – after the development of appropriate mesh generators.

Bibliography

[AD95] Y. Aurégan and C. Depollier. Snoring: Linear stability analysis and in–vitro experiments. *Journal of Sound and Vibration*, 188(1):39–54, 1995.

[ADPS83] B.F. Armaly, F. Durst, J.C.F. Pereira, and B. Schonung. Experimental and theoretical investigation of backward-facing step flow. *Journal of Fluid Mechanics*, 127:473–496, 1983.

[AIA98] AIAA. Guide for the verification and validation of computer fluid dynamics simulations. G-077-1998, 1998.

[Bar96] T.J. Barber. The role of code validation and certification in the design environment. In *27th AIAA Fluid Dynamics Conference, AIAA Paper No.96-2033*, New Orleans, LA, 1996.

[Bat96] Klaus-Jürgen Bathe. *Finite Element Procedures*. Prentice-Hall, Inc., 1996.

[BC97] E. Barragy and G.F. Carey. Stream function-vorticity driven cavity solution using p finite elements. *Computers and Fluids*, 26(5):453–468, 1997.

[Ben60] T.B. Benjamin. Effect of the flexible surface on hydrodynamic stability. *Journal of Fluid Mechanics*, 6:513–532, 1960.

[Ben63] T.B. Benjamin. The treefold classification of unstable disturbances in flexible surface bounding inviscid flows. *Journal of Fluid Mechanics*, 16:436–450, 1963.

[Ber82] C. D. Bertram. Technical note: Two modes of instability in a thick-walled collapsible tube conveying a flow. *Journal of Biomechanics*, Vol.15, No.3:223–224, 1982.

[Ber86] C. D. Bertram. Unstable equilibrium behaviour in collapsible tubes. *Journal of Biomechanics*, Vol.19, No.1:61–69, 1986.

[BH77] D.M. Bushnell and J.N. Heffner. Effect of compliant wall motion on turbulent boundary layers. *Physics of Fluids*, 20:S31–S48, 1977.

[BMD99] D.A. Berry, J.B. Moon, and Kuehn D.P. A finite element model of the soft palate. *Cleft Palate - Craniofacial Journal*, 36(3), 1999.

[Bon99] A. Bonfiglioli. Computational examples: 2D lid-driven cavity flow and backward facing step. http://www.unibas.it/utenti/bonfiglioli/www.html, 1999.

[BP82] C. D. Bertram and T. J. Pedley. A mathematical model of unsteady collapsible tube behaviour. *Journal of Biomechanics*, Vol.15, No.1:39–50, 1982.

[BR90] C. D. Bertram and C. J. Raymond. Mapping of instabilities for flow through collapsible tubes of differing length. *Journal of Fluids and Structures*, 4:125–153, 1990.

[Bra97] D. Braess. *Finite elements, Theory, fast solvers and applications in solid mechanics*. Cambridge University Press, Cambridge, 1997.

[Bro67] R.S. Brodkey. *The Phenomena of Fluid Motions*. Dover Publications, Inc., New York, 1967.

[BZH86] I.G. Brown, N. Zamel, and V. Hoffstein. Pharyngeal cross sectional area in normal men and women. *Journal of Applied Physiology*, 61:890–895, 1986.

[CG85] P.W. Carpenter and A.D. Garrad. The hydrodynamic stability of flow over Kramer-type compliant surfaces; Part 1: Tollmien-Schlichting instabilities. *Journal of Fluid Mechanics*, 155:465–510, 1985.

[CG86] P.W. Carpenter and A.D. Garrad. The hydrodynamic stability of flow over kramer-type compliant surfaces; Part 2: Flow-induced surface instabilities. *Journal of Fluid Mechanics*, 170:199–232, 1986.

[CGW83] P.W. Carpenter, M. Gaster, and G.J.K. Willis. A numerical investigation into boundary layer instability on compliant surfaces. In *Numarical Methods in Laminar and Turbulent Flow*, pages 166–172, Pineridge, Swansea, UK, 1983.

[Che72] R.T. Cheng. Numerical solution of the Navier-Stokes equations by the finite element method. *Physics of Fluids*, 15(12):2098–2105, 1972.

[CLD91] P.W. Carpenter, A.D. Lucey, and A.E. Dixon. The optimisation of compliant walls for drag reduction. *Recent Developments in Turbulence Management*, pages 195–221, 1991. (Printed in the Netherlands).

[CSvS86] C. Cuvelier, A. Segal, and A.A. van Steenhoven. *Finite element methods and Navier-Stokes equations.* Kluwer Academic Publishing Group, D. Reidel Publishing Company, Holland, 1986.

[dAl97] Groupe d'utilisateurs ANSYS l'EPFL. 2D Poiseuille flow analysis. http://dmtwww.epfl.ch/ims/micsys/ANSYS/tp/flopois.html, 1997.

[Dav67] R.T. Davis. Laminar incompressible flow past a semi-infinite flat plate. *Journal of Fluid Mechanics*, 27(4):691–704, 1967.

[DHV+99] K. Dedouch, J. Horácek, T. Vampola, M. Vohradnik, and Svec J. Finite element model of supraglottal vocal tract with consideration of wall impedance. In *3rd International Conference , Engineering Aero-hydroelasticity*, August 1999.

[Din77] A. Dinkelacker. On the problem of drag reduction by means of compliant walls. AGARD Report # 654, (Special Course on Concepts for Drag Reduction), 1977.

[DLC94a] A.E. Dixon, A.D. Lucey, and P.W. Carpenter. Optimisation of viscoelastic compliant walls for transition delay. *AIAA Journal*, 32:256–267, 1994.

[DLC94b] A.E. Dixon, A.D. Lucey, and P.W. Carpenter. Optimization of viscoelastic compliant walls for transition delay. *AIAA Journal*, 32:256–267, 1994.

[DW71] S.C.R. Dennis and J.D. Walsh. Numerical solutions for steady symmetric viscous flow past a parabolic cylinder. *Journal of Fluid Mechanics*, 50(4):801–814, 1971.

[Dyk82] Milton Van Dyke. *An Album of Fluid Motion.* The Parabolic Press, Stanford, California, 1982.

[et.76] Kawahara et.al. Steady and unsteady finite element analysis of incompressible viscous fluids. *International Journal of Numerical Methods in Engineering*, 10:437–456, 1976.

[Fre95] C. J. Freitas. Perspective: Selected benchmarks from commercial CFD codes. *Transactions of the ASME*, 117:208–218, 1995.

[Fun93] Y.C. Fung. *Biomechanics, Mechanical Properties of Living Tissues.* Springer-Verlag, New York, second edition, 1993.

[Fun97] Y.C. Fung. *Biomechanics, Circulation.* Springer-Verlag, New York, second edition, 1997.

[FWBA75] M.C. Fischer, L.M. Weinstein, D.M. Buschnell, and R.L. Ash. Compliant-wall turbulent skin-friction-reduction research. AIAA Paper 75-833, 1975.

[Gar90] D.K. Gartling. A test problem for outflow boundary conditions – flow over a backward-facing step. *International Journal for Numerical Methods in Fluids*, 11:953–967, 1990.

[Gas62] M. Gaster. A note on the relation between temporally-increasing and spatially-increasing disturbances in hydrodynamic stability. *Journal of Fluid Mechanics*, 14:222–224, 1962.

[Gas65] M. Gaster. On the generation of spatially growing waves in a boundary layer. *Journal of Fluid Mechanics*, 22:433–441, 1965.

[Gas87] M. Gaster. Is the dolphin a red herring? *In Proc. IUTAM Symposium on Turbulence Management & Relaminarisation, Bangalore, India*, pages 285–304, 1987.

[Gev99] B. Geveci. *Flow Induced Nonlinear Vibration of Rectangular Plates.* PhD thesis, Lehigh University, 1999.

[GJ93] N. Gavriely and O. Jensen. Theory and measurement of snores. *Journal of Applied Physiology*, 74(6):2828–2837, 1993.

[GP00] C.Q. Guo and M.P. Paidoussis. Stability of rectangular plates with free side-edges in two-dimensional inviscid channel flow. *Journal of Applied Mechanics*, 67:171–176, 2000.

[Gra36] J. Gray. Studies in animal locomotion. VI The propulsive power of the dolphin. *Journal of Experimental Biology*, 13:192–199, 1936.

[Gra77] H. Gray. *Gray's Anatomy.* Gramercy Books, New York, a revised American from the Fifteenth English edition, 1977.

[GS00] P.M. Gresho and P.M. Sani. *Incompressible Flow and the Finite Element Method, Volume One, Advection-Diffusion.* John Wiley and Sons, Ltd., New York, 2000.

[Gun89] Max D. Gunzburger. *Finite element methods for viscous incompressible flows, A Guide to Theory, Practice, and Algorithms, a volume in Computer Science and Scientific Computing.* Academic Press Inc., London, 1989.

[Gyo67] D. Gyorgyfalvy. Possibilities of drag reduction by the use of flexible skin. *Journal of Aircraft*, 4:186–192, February 1967.

[HFW99] L. Huang and J.E. Ffowcs Williams. Neuromechanical interaction in human snoring and upper airway obstruction. *Journal of Applied Physiology*, 86:1759–1763, 1999.

[HH74] R.J. Hansen and D.L. Hunston. An experimental study of turbulent flows over compliant surfaces. *Journal of Sound and Vibration*, 34(3):297–308, 1974.

[HH83] R.J. Hansen and D.L. Hunston. Fluid-property effects on flow-generated waves on a compliant surface. *Journal of Fluid Mechanics*, 133:161–177, 1983.

[HH84] A. Halim and M. Hafez. *Calculation of separation bubbles using boundary layer type equations, Computational Methods in Viscous Flows, Vol.3.* Pineridge, Swansea, 1984.

[Hua95] L. Huang. Flutter of cantilevered plates in axial flow. *Journal of Fluids an Structures*, 9:127–147, 1995.

[Hua98] L. Huang. Reversal of the Bernoulli effect and channel flutter. *Journal of Fluids an Structures*, 12:131–151, 1998.

[Hub00] G. Huber. Swimming in flatsea. *Nature*, 408(6814):777–778, 2000.

[KDO76] A. Kornecki, E.H. Dowell, and J. O'Brien. On the aeroelastic instability of two-dimensional panels in uniform incompressible flow. *Journal of Sound and Vibration*, 47(2):163–178, 1976.

[Kra57] M.O. Kramer. Boundary-layer stabilization by distributed damping. *Journal of the Aeronautical Sciences*, 24:459, June 1957.

[Kra60a] M.O. Kramer. Boundary-layer stabilization by distributed damping. *Journal of the Aero/Space Sciences*, 27:69, January 1960.

[Kra60b] M.O. Kramer. Boundary-layer stabilization by distributed damping. *Journal of the American Society of Naval Engineers*, 72:25–33, February 1960.

[Kra62] M.O. Kramer. Boundary-layer stabilization by distributed damping. *Journal of the American Society of Naval Engineers*, 74:341–348, May 1962.

[KT92] G.E. Karniadakis and G.S. Triantafyllou. Three-dimensional dynamics and transition to turbulence in the wake of bluff objects. *Journal of Fluid Dynamics*, 238:1–30, 1992.

[Kum00] V. Kumaran. Classification of instabilities in the flow past flexible surfaces. *Current Science*, 79(6):766–773, September 2000.

[Lan62] M.T. Landahl. On the stability of a laminar incompressible boundary layer over a flexible surface. *Journal of Fluid Mechanics*, 13:609–632, 1962.

[LCCY97] A.D. Lucey, G.J. Cafolla, P.W. Carpenter, and M. Yang. The nonlinear hydroelastic behaviour of flexible walls. *Journal of Fluids and Structures*, 11:717–744, 1997.

[LK65] M.T. Landahl and R.E. Kaplan. Effect of compliant walls on boundary layer stability and transition. AGAR Dougraph 97-1-353, 1965.

[LP95] X. Y. Luo and T. J. Pedley. A numerical simulation of steady flow in a 2-D collapsible channel. *Journal of Fluids and Structures*, 9:149–174, 1995.

[LP96] X. Y. Luo and T. J. Pedley. A numerical simulation of unsteady flow in a two-dimensional collapsible channel. *Journal of Fluid Mechanics*, 314:191–225, 1996.

[LP98] X. Y. Luo and T. J. Pedley. The effect of wall inertia on flow in a two-dimensional collapsible channel. *Journal of Fluid Mechanics*, 363:253–280, 1998.

[LP00] X. Y. Luo and T. J. Pedley. Multiple solutions and flow limitation in collapsible channel flows. *Journal of Fluid Mechanics*, 420:301–324, 2000.

[Luc89] A.D. Lucey. *Hydroelastic Instability of Flexible Surfaces.* PhD thesis, University of Exeter, 1989.

[Luc98] A.D. Lucey. The excitation of waves on a flexible panel in a uniform flow. *Phil. Trans. R. Soc. Lond. A*, 356:2999–3039, 1998.

[McK94] M.D. McKay. *Aspects of Modeling Uncertainty and Prediction.* Los Alamos National Laboratories, Los Alamos, NM, 1994.

[Meh91] U.B. Mehta. Some aspects of uncertainty in computational fluid dynamics results. *Journal of Fluid Engineering*, 113:538–543, 1991.

[Meh96] U.B. Mehta. Guide to credible computer simulations of fluid flows. *Journal of Propulsion and Power*, 12(5):940–948, 1996.

[Mof64] H.K. Moffatt. Viscous and resistive eddies near a sharp corner. *Journal of Fluid Mechanics*, 18:1–18, 1964.

[MSOE96] Defense Modeling, Office of the Director of Defense Research Simulation Office, and Engineering. Verification, Validation, and Accreditation (VV&A) Recommended Practices Guide. www.dmso.mill/docslib, October 1996.

[NO90] W.W. Nichols and M.F. O'Rourke. *McDonald's Blood Flow in Arteries, Theoretical, Experimental and Clinical Principles.* Lea & Febier, London, third edition edition, 1990.

[Now63] W. Nowacki. *Dynamics of Elastic Systems.* Chapman & Hall Ltd., London, 1963.

[Obe94] W.L. Oberkampf. Proposed framework for computational fluid dynamics code calibration/validation. In *18th AIAA Aerospace Ground Testing Conference, AIAA Paper No.94-2540*, Colorado Springs, CO, 1994.

[otMHBL96] The Editors of the Market House Book Ltd. *The Bantam Medical Dictionary.* Bantam Books, New York, second revised edition, 1996.

[PA67] F. Pan and A. Acrivos. Steady flow in rectangular cavities. *Journal of Fluid Mechanics*, 28:643, 1967.

[Pan84] R.L. Panton. *Incompressible Flow.* Wiley, New York, 1984.

[Par99] M. Paraschivoiu. Mie 1210s computational fluid mechanics and heat transfer. Lecture Notes, University of Toronto, 1999.

[PHS+00] J. Penrose, D. R. Hose, C. J. Staples, I. S. Hamill, I. P. Jones, and D. Sweeney. Fluid structure interactions: coupling of CFD and FE. *CAD-FEM Users' Meeting, Internationale FEM-Technologietage*, September 2000.

[PN92] R.J. Pierce and M. Naughton. Sleep related breathing disorders. *Australian Family Physician*, 21:397–405, 1992.

[Pur62] F.W. Puryear. Bounadry layer control - drag reduction by use of compliant coatings. *NOTS Report No. 1668*, 1962.

[PW99] R.J. Pierce and C.J. Worsnop. Upper airway function and dysfunction in respiration. *Clinical and Experimental Pharmacology and Physiology*, 26:1–10, 1999.

[QTV98] A. Quarteroni, M. Tuveri, and A. Veneziani. Computational vascular fluid dynamics: Problems, models and methods. Technical Report Report EPFL/DMA 11.98, Computing and Visualisation in Science, 1998.

[Red93] J.N. Reddy. *An introduction to the Finite Element Method.* McGraw Hill International Editions, second edition, 1993.

[Red98] B.D. Reddy. *Introductory functional analysis, with applications to boundary value problems and finite elements.* Texts in Applied Mathematics, 27. Springer, 1998.

[RG94] J. N. Reddy and D. K. Gartling. *The Finite Element Method in Heat Transfer and Fluid Dynamics.* CRC Press, Inc., 1994.

[RL96] C.F. Ryan and L.L. Love. Mechanical properties of the velopharynx in obese patients with obstructive sleep apnoea. *Am. J. Respir. Crit. Care Med.*, 154:806–812, 1996.

[Roa97] P.J. Roache. Qualification of uncertainty in computational fluid dynamics. *Annual Review of Fluid Mechanics*, pages 126–160, 1997.

[Roa98] P.J. Roache. *Verification and Validation in Computational Science and Engineering.* Hermosa Publishers, Albuquerque, NM, 1998.

[RS94] Z. Rek and L. Skerget. Boundary element method for steady 2D high-Reynolds number flow. *International Journal for Numerical Methods in Fluids*, 19:343–361, 1994.

[RW94] A. Ramage and A.J. Wathen. Iterative solution techniques for the Stokes and Navier-Stokes equations. *International Journal for Numerical Methods in Fluids*, 19:67–83, 1994.

[SH71] R.W. Spann and R.E. Hyatt. Factors affecting upper airways resistance in conscious man. *Journal of Applied Physiology*, 31:708–712, 1971.

[She90] F.S. Sherman. *Viscous Flow - International Edition.* McGraw-Hill Book Co., Singapore, 1990.

[SK83] F. Schreiber and H.B. Keller. Driven cavity flows by efficient numerical techniques. *Journal of Computational Physics*, 49(2):310–333, 1983.

[SOS82] K.P. Strohl, C.F. O'Cain, and A.S. Slutsky. Alae nasi activation and nasal resistance in healthy subjects. *Journal of Applied Physisology*, 52:1432–1437, 1982.

[SP00] C. Staples and J. Penrose. Fluid-structure interactions: coupling of CFD and FE stress analysis. *CFX Update*, 19:10–11, 2000. ISSN 0965-1608.

[SSF93] Ann Stalheim-Smith and Greg K. Fitch. *Understanding human anatomy and physiology.* West Publishing Company, 1993.

[TBLE91] A. Tully, A. Brancatisano, S.H. Loring, and L.A. Engel. Influence of porterior cricoarytenoid muscle activity on pressure-flow relationship of the larynx. *Journal of Applied Physiology*, 70:2252–2258, 1991.

[Tei97] C. M. Teixeira. Digital physics simulation of lid-driven cavity flow. *International Journal of Modern Physics C*, 8(4):685–696, 1997.

[TH73] C. Taylor and P. Hood. A numerical solution of the Navier-Stokes equations using the finite element technique. *Computers and Fluids*, 1:73–100, 1973.

[Wan91] C.Y. Wang. Exact solutions of the steady-state Navier-Stokes equations. *in Annual Review of Fluid Mechanics*, pages 159–177, 1991.

[Whi95] D.P. White. Pathophysiology of Obstructive Sleep Apnoea. *Thorax*, 50:797–804, 1995.

[Wil86] G.J.K. Willis. *Hydrodynamic Stability of Boundary Layer Over Compliant Surfaces.* PhD thesis, University of Exeter, 1986.

[WLCZ85] D.P. White, R.M. Lombard, R.J. Cadieux, and C.W. Zwillich. Pharyngeal resistance in normal humans: Influence of gender, age and obesity. *Journal of Applied Physiology*, 58:365–371, 1985.

[WM93] D.P. White and W.S. Mezzanotte. Neuromascular compensation in the human upper airway. *Sleep*, 16:90–92, 1993.

[Won77] H. Y. Wong. *A Handbook of Essential Formulae and Data on Heat Transfer for Engineers, First Edition.* Longman Group Ltd., London, 1977.

[WZD97] J.D.A. Walker, C.N. Zhikarev, and T. Delph. Induced oscillations of a finite plate. *AIAA Journal of Guidance, Control and Dynamics*, 20(6):1172–1180, 1997. also in:AIAA 97-0578.

[YYT00a] N. Yamaguchi, K. Yokota, and Y. Tsujimoto. Flutter limit and behaviors of a flexible thin sheet in high-speed flow – I: Analytical method for prediction of the sheet behavior. *Journal of Fluid Engineering*, 122:65–73, 2000.

[YYT00b] N. Yamaguchi, K. Yokota, and Y. Tsujimoto. Flutter limit and behaviors of a flexible thin sheet in high-speed flow – II: Experimental results and predicted behaviors for low mass ratios. *Journal of Fluid Engineering*, 122:74–33, 2000.

[ZCLS00] J. Zhang, S. Childress, A. Libchaber, and M. Shelley. Flexible filaments in a flowing soap film as a model for one-dimensional flags in a two-dimensional wind. *Nature*, 408(6814):835–839, 2000.

Appendices

Appendix A

A brief overview of blood flow in arteries

Another possible application of a numerical code – such as the one described in this thesis – is the computational modeling of blood flow in arteries. However, it should be noted that this summary is for illustration purposes only and it is not explored further in this thesis. It is outlined here to demonstrate the applicability of such a code under different initial, boundary and geometric conditions. For example, fluid dynamics applications using different biofluids (i.e., blood instead of air) and wall boundary conditions (i.e., arterial walls with continuous compliance, which may or may not contain local blood flow blockages). Blood flow in arteries belongs to the field of haemodynamics [NO90] [Fun97] [Fun93].

It can be characterized by the combination of three key phenomena. It is an internal laminar (or sometimes turbulent) flow – with or without flow blockages – having compliant arterial wall boundaries, where the blood flow is pulsatile in nature.

it Laminar Poiseuille flow is used as a classical picture to explain the detailed characteristics of flowing blood. In this, the flow is compared to fully developed viscous flow in a long, circular pipe with a constant flow rate and a parabolic velocity profile. The concept of Poiseuille flow, applies only under laminar, steady-state flow conditions. However, flow in the larger arteries

is, in general, not a Poiseuille type flow. The primary reason for this is the unsteady nature of the flow, with the possibility of transition or at least transitory bursts of turbulence in the aorta. The branching, asymmetric nature of the vascular geometry means that the flow will be characterized fluid mechanically by entry phenomena, asymmetries in the velocity patterns, complicated secondary motions and even flow separation, all of which are far more difficult to analyze than simple steady-state fully developed Poiseuille flow. For artery specific flow and geometry the entrance effect, characterized by the entry length, is at least 150 cm, which is far greater than the length of the aorta. The flow in the aorta thus cannot be characterized as fully developed. In fact, all the larger arteries of the circulatory system, including the epicardial coronary vessel, are subject to entrance effects. In the ascending aorta of large mammals viscous effects of the entrance region are confined to a thin-walled boundary layer. The core flow behaves in an inviscid manner, therefore, the flow is characterized as *largely inviscid.* On the other hand, in the case of fully viscous flow the flow is skewed towards the outer wall. An example for this is the flow in the left common coronary artery. In coronary arteries the Reynolds-numbers are much lower, the viscous effects are more dominant and the flow is laminar. The velocity profile in many regions will be more like a parabolic Poiseuille flow, except that there will be skewing of this profile due to vessel curvature and branching. Significant entrance effects may result in the blunting of the velocity profiles.

Associated with the flow in a bifurcation or curvature is the secondary motion induced by the curved path the fluid must follow. Secondary flows are characterized by a swirling, helical component superimposed on the main streamwise velocity along the tube axis. The most familiar example is that of steady flow in a pipe, a relevant example, since the arch of the aorta is in effect a pipe bend. Secondary motions are also produced by branching, where, just as in a curved pipe, the curvature associated with the change in flow direction is accompanied by a centrifugal pressure gradient. Although there have been numerous fluid-dynamic studies of secondary flow phenomena, instrumentation limitations have prevented any in vivo observa-

tions.

An additional complication introduced by the geometry of the arterial system in flow separation from and reattachment to the wall, causing recirculation zones. This phenomena in pulsatile flows is extremely complex. The recirculation region is unsteady and the separation and reattachment points, if present, can change location or even disappear and then reappear as the flow pulses. The likelihood of separation occurring depends on how sharp a turn the flow must negotiate for the first location and the amount of flow drawn away through the branch for the second location. Even if flow separation does not occur, both these regions would be characterized as generally low in shear, followed by a rapid increase in shear to produce a locally elevated wall shear stress.

The issues addressed above need to be accounted for with suitable numerical models. Therefore, significant modifications to the current model are required to solve problems related to the blood flow phenomena.

Appendix B

Lung volumes, capacities and pressures

Fluid dynamic conditions in the upper airway are influenced by inlet and outlet conditions. These change through the process of breathing. Flow through the upper arway is pressure driven. In any discussion of breathing the following two pressures are important [SSF93]:

1. atmospheric (barametric) pressure
2. intra-alveolar (intra pulmonary pressure)

In effect the intra-alveolar pressure is influenced by the intrapleural pressure. The two "sides" of the continuous pleural membrane, the parietal pleura and the visceal pleura, are attached to each other because of the fluid between them. (See Figure B.1) This fluid is located within the pleural cavity. The natural recoil tendency of the lungs tends to pull the parietal and visceal pleura apart and thereby causes intrapleural pressure to be slightly less than atmospheric pressure (p_{atm}).

The flow rate through the upper airway is dependent on lung volumes and capacities. Figure B.2 illustrates the definitions given below, which are associated with these volumes and capacities:

Figure B.1: Simplified representation of the lungs [SSF93]

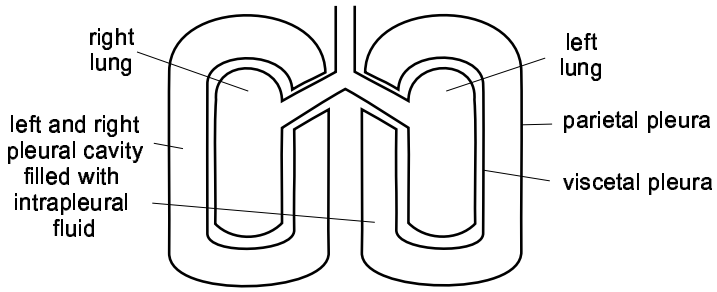
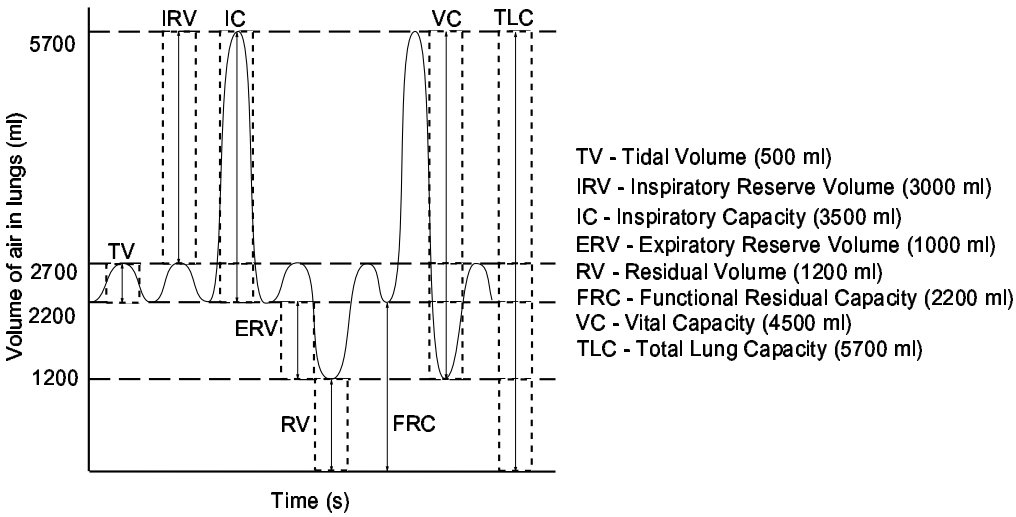


Figure B.2: Lung volumes and capacities [SSF93]



- `Tidal Volume (TV)`: the volume of air that moves into and out of the lungs at each breath during normal, quiet breathing. The tidal volume is typically about 500 ml. With exercise, the tidal volume increases dramatically, to meet the changed metabolic demand of the cell.

- `Inspiratory Reverse Volume (IRV)`: during the deepest possible inspiration, the added volume of air that can be moved into the lungs over and above the tidal volume (TV).

- `Inspiratory Capacity (IC)`: The maximum amount of air that can be inspired after a normal expiration (IC=TV+IRV).

- `Expiratory Reverse Volume (ERV)`: during normal expiration the volume of air that can be moved out of the lungs beyond the end of tidal breth (avg. approx. 1000 ml).

- `Residual Volume (RV)`: a maximal expiration does not completely empty the lungs of air. RV is the volume of air remaining in the lungs after the maximal expiration. It averages 1200 ml and ensures that the lungs do not collapse upon forceful expiration.

- `Functional Residual Capacity (FRC)`: the amount of air remaining after a normal expiration (FRC= ERV+RV). The air represented by FRC ensures that gas exchange in the lungs occurs continuously, even at the end of an expiration.

- `Vital Capacity (VC)`: all of the lung volumes except the residual volume (avg. approx. 4500 ml). (VC=IRV+TV+ERV)

- `Total Lung Capacity (TLC)`: what the lungs can hold (TLC=IRV+TV+ERV+RV).

These volumes, in conjunction with the upper airway geometries, can be used to calculate relevant flow rates and flow velocity boundary conditions in numerical simulations.

Appendix C

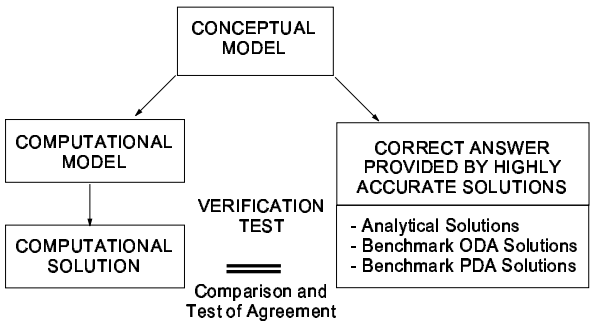
Verification and validation of computer codes

One of the primary factors in the rate of growth of CFD as a research and engineering tool in the future will be the level of credibility that can be developed in the simulations produced. Key to building this credibility is the development of commonly accepted and applied verification and validation (V&V) terminology and methodology. The information and definitions provided in this Appendix were extracted from the document entitled *Guide for the Verification and Validation of Computer Fluid Dynamics Simulations* [AIA98].

Verification is the process of determining the accuracy of a given computational solution; that is, has the problem been solved correctly? The fundamental strategy of verification is the identification and quantification of error in the computational solution (see Figure C.1). Verification activities are primarily performed early in the development cycle of a CFD code. However, these activities do need to be confirmed when the code is subsequently modified or enhanced.

Validation is the process of assessing the degree to which the computational simulation represents the real world; that is, has the correct problem been solved? The fundamental strategy of validation is the identification and quantification of error and uncertainty in the mathematical

Figure C.1: Verification process [AIA98]



model of the physics and computational solution. (see Figure C.2)

There are several validation methods available. Figure C.3 shows a building block approach, which is used in the present research. Each phase of the process represents a different level of flow physics coupling and geometrical complexity. A summary is show in Table C.1. The complete system consists of the actual system for which a validated CFD tool is needed. Thus, by definition, all the geometric and flow physics effects occur simultaneously; commonly, the complete system includes multidisciplinary physical phenomena. Data are measured on the engineering hardware under realistic operating conditions. These measurements, however, are very limited. Exact test conditions (for example, initial and boundary conditions) are hard to quantify, and the data generally have a fairly high degree of uncertainty.

Figure C.4 clearly shows the meaning of V&V and their relationship to one another. It identifies two types of models: a conceptual model and a computerized model. The conceptual model is composed of all the information, mathematical modeling data, and mathematical equations that describe the physical system or process of interest. The conceptual model is produced by analysis and observations of the physical system. In CFD, the conceptual model is dominated by the partial differential equations (PDEs) for conservation equations of mass, momentum and energy. The computerized model is an operational computer program that

Figure C.2: Validation process [AIA98]

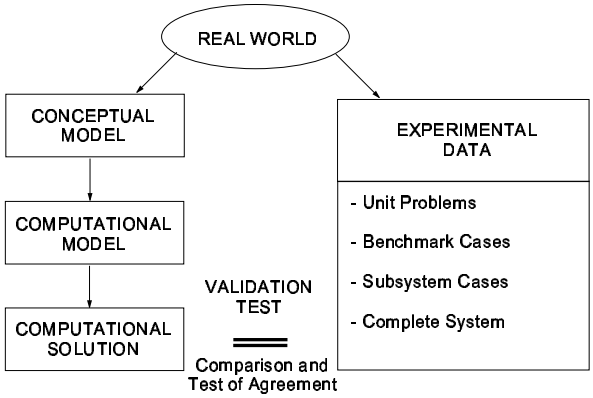


Figure C.3: Validation phases [AIA98]

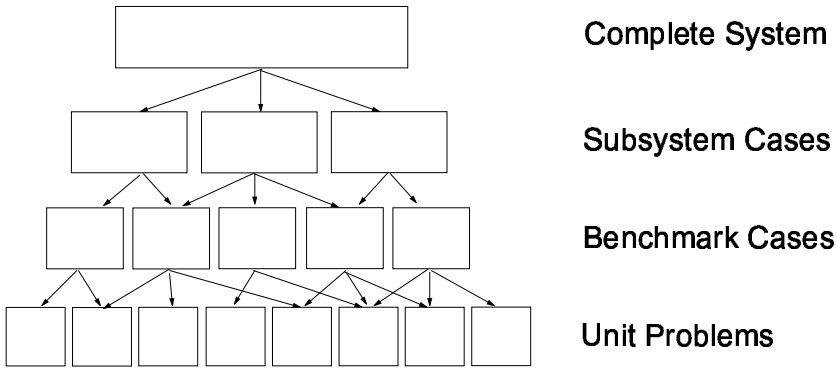
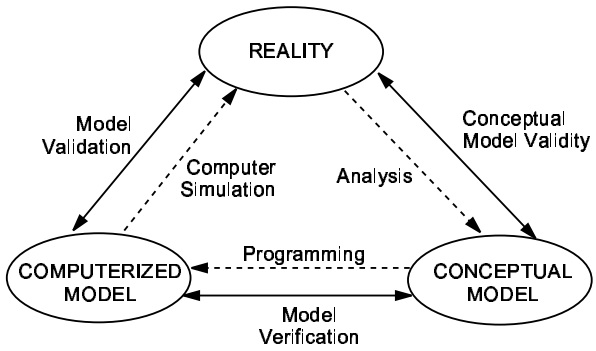


Table C.1: Characteristics of Validation [AIA98] Phases

Complete System	Actual System Hardware Complete Flow Physics All Relevant Flow Features	Limited Experimental Data Most Initial Conditions and Boundary Conditions Unknown
Subsystem Cases	Subsystem or Component Hardware Moderately Complex Flow Physics Multiple Relevant Flow Features	Large Experimental Uncertainty Some Initial Conditions and Boundary Conditions Measured
Benchmark Cases	Special Hardware Fabricated Two Elements of Complex Flow Physics Two Relevant Flow Features	Moderate Experimental Uncertainty Most Initial Conditions and Boundary Conditions Measured
Unit Problems	Simple Geometry Hardware Fabricated One Element of Complex Flow Physics One Relevant Flow Feature	Low Experimental Uncertainty All Initial Conditions and Boundary Conditions Measured

Figure C.4: Phases of modeling and simulation [AIA98]



implements a conceptual model. Modern terminology refers to the computerized model as the computer model or code.

Another important factor affecting the credibility of predictions is the level of complexity involved. There are three aspects of complexity in modeling and simulation that should be addressed [Obe94] [Bar96]:

1. the complexity of the physical model
2. the complexity of the model representing the physics
3. the level of prediction difficulty of an output quantity from the simulation

Regarding the complexity of the physical process, fluid dynamics provides an extraordinarily wide range of complexity. The following categories provide one way of viewing the complexity of the physical process:

- Spatial dimensionality
- Temporal nature
- Geometry
- Flow physics

As one deals with these different aspects in a simulation, the credibility of the predictions is directly affected. If the simulation is restricted to a specific class of problems for which the CFD model has been validated, confidence in the accuracy of the solution is clearly enhanced.

There are four predominant sources of error in CFD simulations:

- insufficient spatial discretization convergence,
- insufficient temporal discretization convergence,
- lack of iterative convergence and
- computer programming.

The procedures for estimating spatial (i.e., grid) and temporal convergence are similar. Procedures for estimating iterative convergence include techniques for consistency checks on the solution. The programming errors are not addressed. An extensive description of verification methodology and procedures is given by Roache [Roa98].

There is no fixed requirement of accuracy that is applicable to all CFD simulations. The accuracy level required of simulations depends on the purposes for which the simulations are to be used. Not all simulations need to demonstrate high accuracy as long as the error and uncertainty of the simulations can be estimated. In CFD simulations of complex engineering system, the accuracy level is influenced by such factors as cost, schedule and risk of the failure of the system.

Other fundamental terms include uncertainty, error, prediction, and calibration. Further details on these can be found in the appropriate references. In simulations involving complex flow physics or multidisciplinary engineering systems, strict validation procedures commonly become impractical. For example, when all of the important physical modeling parameters are not known *a priory*, some of the parameters are considered adjustable. When grid-resolved

solutions are not attainable because of the computer resources needed for the simulation, adjustments must be made to improve agreement with the experimental data. When these types of activities occur, the term *calibration* more appropriately describes the process than does validation.

- `Uncertainty`: A potential deficiency in any phase or activity of the modeling process that is due to lack of knowledge.
- `Error`: A recognizable deficiency in any phase or activity of modeling and simulation that is not due to lack of knowledge.

Uncertainty and error can be considered as the broad categories that are normally associated with loss in accuracy in modeling and simulation. Some errors, like computer round-off and iterative errors are well understood. On the other hand, some errors, such as the numerical error in the discrete solution of PDEs with singularities or discontinuities, are not well understood. Other shortcomings in modeling and simulations are associated with uncertainties rather than errors. Examples are the uncertainty in the surface roughness in the simulation of flow over a turbine blade and the uncertainty in the validity of a turbulence model.

In CFD literature the terms uncertainty and error have commonly been used interchangeably [Meh91] [Roa97]. Uncertainty is defined as a potential deficiency, meaning that it may or may not occur. The second feature of uncertainty is that its fundamental cause is lack of knowledge. It can be addressed through: a sensitivity analysis and an uncertainty analysis [McK94] [Meh96]. A sensitivity analysis is composed of multiple simulations from a code to determine the effect of the variation of some component of the model, such as an input parameter or modeling assumptions, on certain output quantities. However, this type of analysis do not normally deal with the interaction of various uncertainty sources or the relative level of confidence in variations. An example of the uncertainty analysis is a Monte Carlo simulation.

An error is identifiable or knowable upon examination. The error can be acknowledged or unacknowledged. Examples of acknowledged errors are round-off errors in a computer, physical approximations made to simplify the modeling of a physical process and a specified level of iterative convergence of a numerical scheme. Unacknowledged errors include blunters and mistakes, commonly caused by people (e.g., programming errors).

- `Prediction`: Use of a CFD model to foretell the state of a physical system under conditions for which the CFD model has not been validated.

A prediction refers to the computational simulation of a specific case of interest that is different from cases that have been validated. It goes beyond agreement with previous experimental data in the validation database.

The V&V processes do not directly make claim about the accuracy of predictions [MSOE96]. The accuracy of predictions from a computational model is not guaranteed by V&V processes because of the extraordinary non-uniqueness of the computational model. These processes do not address future usage of the code such as: correctness of the input parameters, accuracy of the new geometry of interest, skill of the code user, appropriateness of the modeling assumptions and the quality of grid generation.

Benchmark ODE solutions are very accurate numerical solutions to special cases of the general CFD model. These ODEs commonly result from simplifying assumptions, such as simplified geometries and assumptions resulting in the formation of similarity variables. An example is the Blasius solution for laminar flow over a flat plate [Pan84] [Wan91].

Benchmark PDE solutions are also very accurate numerical solutions to special cases of either the PDEs or the boundary conditions. Examples of various representative benchmark PDE solutions are the following: incompressible laminar flow over a semi-infinite flat plate [Dav67], incompressible laminar flow over a parabolic plate [DW71], incompressible laminar flow over

a square cavity driven by a moving wall [SK83] [BC97], incompressible laminar flow over an infinite-length circular cylinder [KT92], and incompressible laminar flow over a backward-facing step, with and without heat transfer [Gar90].

Note that as one moves from ODE solutions to PDE solutions, the accuracy of th benchmark solutions clearly becomes more of an issue.

Further information on code verification and validation can be found in the appropriate literature.

Appendix D

BC's for 2D duct flows with unsteady inlet flow conditions

The velocity distribution for an unsteady inlet velocity case is similar to the one described for the steady inlet velocity case, with the addition of a simple periodicity multiplier. The inlet velocity boundary condition can be re-defined through the C multiplier as

$$C = \left(1 + \frac{sin(t)}{2}\right) \tag{D.1}$$

and therefore the inlet velocity in the x-direction can be calculated from

$$u_{1in} = \left(1 + \frac{sin(t)}{2}\right)\left(1 - \left(\frac{y}{Y}\right)^2\right) \tag{D.2}$$

where t is the time series time of the simulation, y is the distance from the centerline and Y is the half height of the duct. This modification results in periodic peak inlet velocities between 0.5 and 1.5.

The average velocity in both the steady and unsteady cases can be derived by integrating the velocity profile

$$A = 2C\int_0^Y \left(1 - \left(\frac{y}{Y}\right)^2\right) = 2C\left[y - \frac{y^3}{3Y^2}\right]_0^Y = 2C\left[Y - \frac{Y}{3}\right] \tag{D.3}$$

the integral calculates from 0 to Y, which is only half of the velocity profile.

From this the average velocity in the x-direction is

$$\bar{u}_1 = \frac{A}{2Y} = \frac{2C}{2Y}\left[Y - \frac{Y}{3}\right] = C\left[1 - \frac{1}{3}\right] = C\frac{2}{3} \tag{D.4}$$

where $2Y$ is the height of the duct.

Appendix E

Practical aspects of the FEM method

E.1 FEM Block Formation Process

The following Appendix describes the block formation process from the practical computer code development point of view. First, the elemental constructions outline the derivation of the elemental stiffness, mass, gradient and convection matrices. In the second subsection the elemental constructions are assembled into global constructions, such as the global system matrix, the forcing term and the convection term.

E.1.1 Elemental constructions

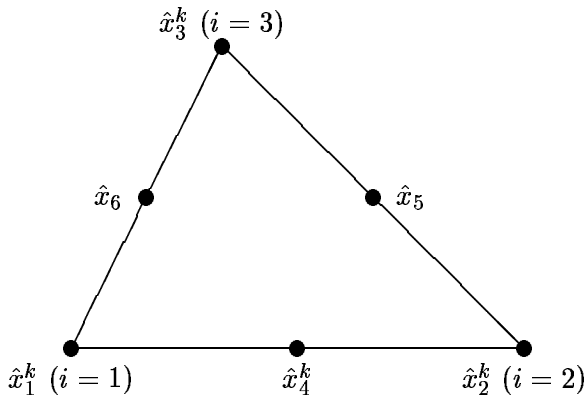
Derivation of the Elemental Stiffness Matrix

This section shows the derivation of the Elemental Stiffness Matrix.

Note that in the subscripts $[i] = i \ mod \ 3$, refers to the modulus, where for example $[4] = 1$. The calculations cycle through the local elements and the use of modulus makes sure that once the maximum number of local elements is reached, it falls back to the first element and so on.

The elemental construction is shown in Figure E.1. For a given element the local nodes are shown, where for a quadratic triangular element the nodes vary between 1 and 6.

Figure E.1: Elemental construction for a triangular element



The area of a given element can be expressed as:

$$area^k = \frac{1}{2}[(\hat{x}^k_{[i+1]} - \hat{x}^k_{[i]}) \times (\hat{x}^k_{[i+2]} - \hat{x}^k_{[i]})] \cdot e_3 \tag{E.1}$$

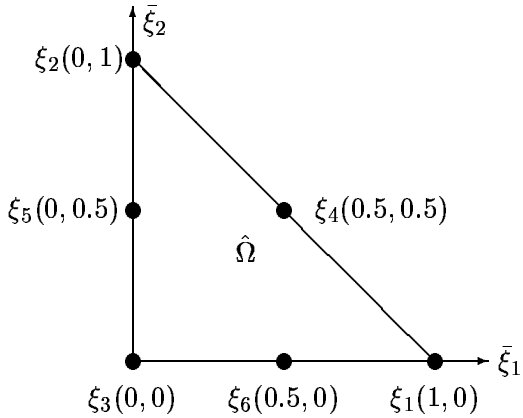
In these calculations i varies from 1 to 3, representing the corner nodes of the triangular element. Also note that e_3 represents the unit vector in the third dimension, due to the cross product of the two vectors.

$$area^k = \frac{1}{2}\begin{bmatrix} 0 & 0 & 1 \\ \hat{x}^k_{[i+1]} - \hat{x}^k_{[i]} & \hat{y}^k_{[i+1]} - \hat{y}^k_{[i]} & 0 \\ \hat{x}^k_{[i+2]} - \hat{x}^k_{[i]} & \hat{y}^k_{[i+2]} - \hat{y}^k_{[i]} & 0 \end{bmatrix} \tag{E.2}$$

For the reference element shown in Figure E.2, a given point ξ_i can be calculated from the following correlation, where the area refers to the element area in the elemental construction:

$$\xi_i = \frac{1}{2}[(\hat{x}^k_{[i+1]} - x) \times (\hat{x}^k_{[i+2]} - x)] \cdot \frac{e_3}{area^k} \tag{E.3}$$

Figure E.2: Elemental nodal basis for a triangular element



In a more general form, ξ can be expressed as:

$$\begin{bmatrix} \xi_1 \\ \xi_2 \end{bmatrix} = \begin{bmatrix} c_1^k \\ c_2^k \end{bmatrix} + \begin{bmatrix} D_{11}^k & D_{12}^k \\ D_{21}^k & D_{22}^k \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \tag{E.4}$$

From this, c_i^k, and the affine mapping derivative matrices (D_{i1}^k and D_{i2}^k) are calculated from the following equations:

$$c_i^k = \frac{1}{2area^k}\{\hat{x}_{[i+1]}^k \hat{y}_{[i+2]}^k - \hat{x}_{[i+2]}^k \hat{y}_{[i+1]}^k\} \tag{E.5}$$

$$D_{i1}^k = \frac{1}{2area^k}\{\hat{y}_{[i+1]}^k - \hat{y}_{[i+2]}^k\} = \left.\frac{\partial \xi_i}{\partial x_1}\right|_{\Omega^k} \tag{E.6}$$

$$D_{i2}^k = \frac{1}{2area^k}\{\hat{x}_{[i+2]}^k - \hat{x}_{[i+1]}^k\} = \left.\frac{\partial \xi_i}{\partial x_2}\right|_{\Omega^k} \tag{E.7}$$

The Elemental Stiffness Matrix is derived using the following steps:

$$\hat{A}_{\alpha\beta}^k = \int_{\Omega^k} \sum_{m=1}^{2} \frac{\partial h_\alpha}{\partial x_m}\frac{\partial h_\beta}{\partial x_m} dx_1 dx_2 \tag{E.8}$$

$$= 2area^k \int_{\hat{\Omega}} \sum_{m=1}^{2} \left(\sum_{n=1}^{2} \frac{\partial h_\alpha}{\partial \xi_n}\frac{\partial \xi_n}{\partial x_m}\right)\left(\sum_{n=1}^{2} \frac{\partial h_\beta}{\partial \xi_n}\frac{\partial \xi_n}{\partial x_m}\right) d\xi_1 d\xi_2 \tag{E.9}$$

$$= 2area^k \int_{\hat{\Omega}} \sum_{m=1}^{2} \left[\frac{\partial h_\alpha}{\partial \xi_1}\frac{\partial \xi_1}{\partial x_m} + \frac{\partial h_\alpha}{\partial \xi_2}\frac{\partial \xi_2}{\partial x_m}\right]\left[\frac{\partial h_\beta}{\partial \xi_1}\frac{\partial \xi_1}{\partial x_m} + \frac{\partial h_\beta}{\partial \xi_2}\frac{\partial \xi_2}{\partial x_m}\right] d\xi_1 d\xi_2 \quad \text{(E.10)}$$

$$= 2area^k \int_{\hat{\Omega}} \sum_{m=1}^{2} \left[\frac{\partial h_\alpha}{\partial \xi_1}D^k_{1m} + \frac{\partial h_\alpha}{\partial \xi_2}D^k_{2m}\right]\left[\frac{\partial h_\beta}{\partial \xi_1}D^k_{1m} + \frac{\partial h_\beta}{\partial \xi_2}D^k_{2m}\right] d\xi_1 d\xi_2 \quad \text{(E.11)}$$

$$= 2area^k \int_{\hat{\Omega}} \sum_{m=1}^{2} \left[\sum_{n=1}^{2}\frac{\partial h_\alpha}{\partial \xi_n}D^k_{nm}\right]\left[\sum_{n'=1}^{2}\frac{\partial h_\beta}{\partial \xi_{n'}}D^k_{n'm}\right] d\xi_1 d\xi_2 \quad \text{(E.12)}$$

$$= 2area^k \sum_{m=1}^{2}\sum_{n=1}^{2}\sum_{n'=1}^{2} D^k_{nm}D^k_{n'm}G_{\alpha\beta nn'} \quad \text{(E.13)}$$

where

$$G_{\alpha\beta nn'} = \int_0^1 \int_0^{1-\xi_2} \frac{\partial h_\alpha}{\partial \xi_n}\frac{\partial h_\beta}{\partial \xi_{n'}} d\xi_1 d\xi_2 \quad \text{(E.14)}$$

The derivation of the $G_{\alpha\beta nn'}$ term is given in Appendix F.

Elemental mass matrix

A MAPLE script is used to calculate the Mass Matrix for Quadratic Triagular Elements. The calculations are performed using the symbolic software MAPLE V 5.1.

The Mass Matrix for Quadratic Triangular and for Linear Elements are defined as:

$$\hat{M}^k_{\alpha\beta} = \int_{\Omega^k} h_\alpha h_\beta dx dy \quad \text{(E.15)}$$

$$= 2\ area^k \int_0^1 \int_0^{1-\xi_2} h_\alpha h_\beta d\xi_1 d\xi_2 \quad \text{(E.16)}$$

where α and β varies from 1 to 6 for quadratic elements, from 1 to 3 for linear elements, and k represents the element number.

Detailed calculations of the mass matrices for quadratic and linear elements are given in Appendix H.

Derivation of the Elemental Gradient Matrix

This sub-section shows the derivation of the elemental gradient matrix ($(\widehat{D}_i)^k_{\alpha\beta}$) used in (the pressure related terms of) the Stokes and Navier-Stokes equations. The correlations for $area^k$, ξ_i, c^k_i, D^k_{i1} and D^k_{i2} (these are different from the affine mapping derivative matrices, $(\widehat{D}_i)^k_{\alpha\beta}$) are calculated from Equations E.1 to E.7.

$(\widehat{D}_i)^k_{\alpha\beta}$ (where $i = 1, 2$ in 2D) can be expressed as:

$$(\widehat{D}_i)^k_{\alpha\beta} = \int_{\Omega^k} h^P_\alpha \frac{\partial h^V_\beta}{\partial x_i}\, dx_1 dx_2 \tag{E.17}$$

from this,

$$\begin{aligned}
(\widehat{D}_i)^k_{\alpha\beta} &= \int_{\Omega^k} h^P_\alpha \frac{\partial h^V_\beta}{\partial x_i} dx_1 dx_2 && \text{(E.18)}\\
&= 2area^k \int_{\hat{\Omega}} h^P_\alpha \left(\sum_{n=1}^{2} \frac{\partial h^V_\beta}{\partial \xi_n} \frac{\partial \xi_n}{\partial x_i} \right) d\xi_1 d\xi_2 && \text{(E.19)}\\
&= 2area^k \int_0^1 \int_0^{1-\xi_2} h^P_\alpha \left[\frac{\partial h^V_\beta}{\partial \xi_1} \frac{\partial \xi_1}{\partial x_i} + \frac{\partial h^V_\beta}{\partial \xi_2} \frac{\partial \xi_2}{\partial x_i} \right] d\xi_1 d\xi_2 && \text{(E.20)}\\
&= 2area^k \int_0^1 \int_0^{1-\xi_2} h^P_\alpha \left[\frac{\partial h^V_\beta}{\partial \xi_1} D^k_{1i} + \frac{\partial h^V_\beta}{\partial \xi_2} D^k_{2i} \right] d\xi_1 d\xi_2 && \text{(E.21)}\\
&= 2area^k \int_0^1 \int_0^{1-\xi_2} h^P_\alpha \left[\sum_{n=1}^{2} \frac{\partial h^V_\beta}{\partial \xi_n} D^k_{ni} \right] d\xi_1 d\xi_2 && \text{(E.22)}\\
&= 2area^k \sum_{n=1}^{2} D^k_{ni} \int_0^1 \int_0^{1-\xi_2} h^P_\alpha \frac{\partial h^V_\beta}{\partial \xi_n} d\xi_1 d\xi_2 && \text{(E.23)}\\
&= 2area^k \sum_{n=1}^{2} D^k_{ni} G^{\widehat{D}}_{\alpha\beta n} && \text{(E.24)}
\end{aligned}$$

where

$$G^{\widehat{D}}_{\alpha\beta n} = \int_0^1 \int_0^{1-\xi_2} h^P_\alpha \frac{\partial h^V_\beta}{\partial \xi_n} d\xi_1 d\xi_2 \tag{E.25}$$

Note that in the above equations $\alpha = 1, ..., 3$, $\beta = 1, ..., 6$ and $(\widehat{D}_1)^k_{\alpha\beta}$ and $(\widehat{D}_2)^k_{\alpha\beta}$ are 3×6 matrices for each element.

Further details on the derivation of the Elemental Gradient Matrix is given in Appendix I.

Elemental convection matrix

The elemental convection matrix, $\left(\widehat{C}^k_{\alpha\delta}\right)^n$, is given as:

$$\left(\widehat{C}^k_{\alpha\delta}\right)^n = 2area^k \sum_{\gamma=1}^{2}\sum_{\beta=1}^{6}\sum_{j=1}^{2}(u^n_j)^k_\beta \; D^k_{\gamma j} \; G_{\alpha\beta\delta\gamma}, \quad (\alpha,\delta) \in \{1,...,6\}^2 \tag{E.26}$$

where $D^k_{\gamma j}$ is the affine mapping derivative matrix (see Equations E.6 and E.7),

$$G_{\alpha\beta\delta\gamma} = \int_{\hat{\Omega}} h_\alpha h_\beta \frac{\partial h_\delta}{\partial \xi_\gamma} \mathrm{d}\xi_1 \mathrm{d}\xi_2 \tag{E.27}$$

and $(u^n_j)^k_\beta$ $(j = 1, 2)$ is the j^{th} component of the velocity field at the n^{th} time step in the k^{th} element at the β^{th} local node $(\beta = 1, ..., 6)$.

A MAPLE script to derive $G_{\alpha\beta\delta\gamma}$ and the derivation results are given in Appendix J.

The elemental convection matrix (6×6) was calculated for each element using the following logic:

zero $(\widehat{C})_{k,\alpha,\delta}$

{For $1 \leq k \leq$ Number of elements (K)

{For $1 \leq \alpha \leq$ (Number of local velocity nodes (=6))

{For $1 \leq \delta \leq$ (Number of local velocity nodes (=6))

{For $1 \leq \gamma \leq$ 2)

{For $1 \leq \beta \leq$ (Number of local velocity nodes (=6))

$$
\begin{aligned}
& i = (g^V)^k_\beta \\
& \{\text{For } 1 \leq j \leq \ 2) \\
& (\widehat{C})_{k,\alpha,\delta} = (\widehat{C})_{k,\alpha,\delta} + (u^n_j)_{i,j} * D_{k,\gamma,j} * G_{\alpha\beta\delta\gamma} \\
\} \quad \} \quad \} \quad \} \quad \} \\
& (\widehat{C})_{k,\alpha,\delta} = 2 \ * \ Area^k \ * \ (\widehat{C})_{k,\alpha,\delta}\}
\end{aligned} \tag{E.28}
$$

The elemental convection vector (6×1) is obtained from

$$\sum_{\delta=1}^{6} \left(\hat{C}^k_{\alpha\delta}\right)^q u^q_{g^k_\delta}$$

using the following logic:

$$
\begin{aligned}
& \text{zero}(\widehat{C}U1\_ELEM) \text{ and } (\widehat{C}U2\_ELEM) \\
& \{\text{For } 1 \leq k \leq \text{Number of elements (K)} \\
& \quad \{\text{For } 1 \leq \alpha \leq \ \text{(Number of local velocity nodes (=6))} \\
& \quad \quad \{\text{For } 1 \leq \delta \leq \ \text{(Number of local velocity nodes (=6))} \\
& \quad \quad \quad i = (g^V)^k_\beta \\
& \quad \quad \quad (\widehat{C}U1\_ELEM)_{k,\alpha} = (\widehat{C}U1\_ELEM)_{k,\alpha} + (\widehat{C})_{k,\alpha,\delta} * u_1 \\
& \quad \quad \quad (\widehat{C}U2\_ELEM)_{k,\alpha} = (\widehat{C}U2\_ELEM)_{k,\alpha} + (\widehat{C})_{k,\alpha,\delta} * u_2 \\
& \} \quad \} \quad \}
\end{aligned} \tag{E.29}
$$

There are two sets of elemental convection vectors, one associated with the u_1 and another with the u_2 velocity components, with vectors for each element. These vectors are used to assemble the global convection term, explained in Section E.1.2.

E.1.2 Global constructions

Assembly of the system matrix

The first step in assembling the system matrix is *assembling the global stiffness matrix* $(\widetilde{A})$ from the elemental stiffness matrices $(\hat{A}^k_{\alpha\beta})$, and the *global mass matrix* $(\hat{M}^k_{\alpha\beta})$ from the elemental

mass matrices ($\hat{M}^k_{\alpha\beta}$). The size of $\widetilde{A}$ and $\widetilde{M}$ are ($\mathcal{M}^{\mathcal{V}} \times \mathcal{M}^{\mathcal{V}}$). The algorithm to assemble $\widetilde{A}$ and $\widetilde{M}$ are given below:

$$
\begin{array}{l}
\{\text{For } 1 \leq k \leq \text{Number of elements (K)} \\
\quad \{\text{For } 1 \leq \alpha \leq \text{ (Number of local velocity nodes (=6))} \\
\quad\quad \{i = (g^V)^k_\alpha \\
\quad\quad\quad \{\text{For } 1 \leq \beta \leq \text{ (Number of local velocity nodes (=6))} \\
\quad\quad\quad\quad \{j = (g^V)^k_\beta \\
\quad\quad\quad\quad (\widetilde{A})_{ij} = (\widetilde{A})_{ij} + (\widehat{A})^k_{\alpha\beta} \\
\quad\quad\quad\quad (\widetilde{M})_{ij} = (\widetilde{M})_{ij} + (\widehat{M})^k_{\alpha\beta} \\
\} \quad \} \quad \} \quad \} \quad \}
\end{array}
\tag{E.30}
$$

Two identical sums of global stiffness and mass matrices (see Equation E.32 below) will form the upper section of the system matrix, with complementing zero matrices. In a way, these two sums of matrices represent the velocity nodes for the x and y coordinate directions.

The second step in assembling the system matrix is *assembling the global gradient matrices* ($\widetilde{D}_1$ and $\widetilde{D}_2$) from the elemental gradient matrices ($(\widehat{D}_i)^k_{\alpha\beta}$). The size of both $\widetilde{D}_1$ and $\widetilde{D}_2$ matrices are ($\mathcal{M}^{\mathcal{P}} \times \mathcal{M}^{\mathcal{V}}$). These global matrices will form the pressure related lower and right side sections of the system matrix (with supplementing zero matrices at the lower right part of the system matrix).

The assembly of $\widetilde{D}_i$ (i=1,2) is performed using the following algorithm (similar to the assembly of the global stiffness matrix). (The algorithm will be given for $\widetilde{D}_1$, while for $\widetilde{D}_2$ the same logic will apply):

$$
\begin{array}{l}
\text{zero } \widetilde{D}_1
\end{array}
$$

$$
\begin{array}{l}
\{\text{For } 1 \leq k \leq \text{Number of elements (K)} \\
\quad \{\text{For } 1 \leq \alpha \leq \text{ (Number of local pressure nodes (=3))} \\
\quad\quad \{i = (g^P)^k_\alpha \\
\quad\quad\quad \{\text{For } 1 \leq \beta \leq \text{ (Number of local velocity nodes (=6))} \\
\quad\quad\quad\quad \{j = (g^V)^k_\beta \\
\quad\quad\quad\quad (\tilde{D}_1)_{mn} = (\tilde{D}_1)_{mn} + (\hat{D}_1)^k_{\alpha\beta} \\
\} \quad \} \quad \} \quad \} \quad \}
\end{array} \tag{E.31}
$$

Once the global stiffness matrix ($\widetilde{A}$) the global mass matrix ($\widetilde{M}$), and the global gradient matrices ($\tilde{D}_1$ and $\tilde{D}_2$) are obtained, the system matrix for the left hand side of the Navier–Stokes equation can be assembled as:

$$
\widetilde{S} = \left[\begin{array}{|c|c|c|}
\hline
\frac{1}{Re}\widetilde{A} + \frac{1}{\Delta t}\widetilde{M} & 0 & -\tilde{D}_1^T \\
\hline
0 & \frac{1}{Re}\widetilde{A} + \frac{1}{\Delta t}\widetilde{M} & -\tilde{D}_2^T \\
\hline
-\tilde{D}_1 & -\tilde{D}_2 & 0 \\
\hline
\end{array}\right] \tag{E.32}
$$

The size of the system matrix is $[(\mathcal{M}^{\mathcal{V}} + \mathcal{M}^{\mathcal{V}} + \mathcal{M}^{\mathcal{P}}) \times (\mathcal{M}^{\mathcal{V}} + \mathcal{M}^{\mathcal{V}} + \mathcal{M}^{\mathcal{P}})]$, where $\mathcal{M}^{\mathcal{V}}$ and $\mathcal{M}^{\mathcal{P}}$) are the number of velocity and pressure nodes, respectively.

Assembly of the forcing vector

In general, the forcing term is given in the following form:

$$
\widetilde{F} = [\widetilde{f} + \widetilde{v}] + \text{Conv.} \tag{E.33}
$$

where "Conv." is the convection term.

$\widetilde{f}$ term

The first component, $\widetilde{f}$, can be written as:

$$\widetilde{f} = \begin{bmatrix} \widetilde{f}_1 \\ \vdots \\ \widetilde{f}_2 \\ \vdots \\ 0 \\ \vdots \end{bmatrix} \tag{E.34}$$

In the calculations $\widetilde{f}$ is first initialized to zero. The boundary conditions are given in Chapter 3.2, specify the velocities along the boundaries.

This boundary condition (e.g., the bottom lid velocity or the duct inlet velocity) will provide the right hand side of the system of equations, once they are assembled. Since the velocity is known for the boundary nodes, the system of equations will not be solved for these corresponding variables. For example, in the case of the lid-driven cavity flow using a tiangular domain, the bottom mask ($mskbot_i$) flags these bottom nodes. By looping through $mskbot_i$, the corresponding columns from the system matrix are moved to the right hand side with a negative sign. (More information on the masks will be given in Section E.2.4) The algorithm to achieve this is presented here using an example of the lid-driven cavity flow case:

zero $\widetilde{f}$

for $i = 1$ to $(\mathcal{M}^{\mathcal{V}} + \mathcal{M}^{\mathcal{V}} + \mathcal{M}^{\mathcal{P}})$

  if $mskbot_{big} = 0$

    $\widetilde{f} = \widetilde{f} - \widetilde{S}\ (all\ rows, i)$

  end

end (E.35)

where $mskbot_{big}$ is a combined $mskbot$ vector for $\widetilde{f}_1$ (reflecting the BC's for $u_1 = 1$ on the

bottom boundary nodes - these are affected by BC's), $\tilde{f}_2$ (reflecting the BC's for $u_2 = 0$ on the bottom boundary nodes - these are not affected by BC's) and 1's (for the pressure related nodes - these are also not affected by BC's) ([*mskbot*; *ones*; *ones*]).

The rows and columns in $\tilde{S}$ and the corresponding nodes in $\tilde{f}$ will be removed later with the help of $mask_i$ as will be explained in the next section for the Direct Solver (see Section E.3).

Note that, in addition to the boundary conditions, before the system is solved, one (reference) pressure node must be removed from both sides. Further on this will be added in the next section (see Section E.3).

Similarly to the wedge shaped domain, the $\tilde{f}$ term can be assembled for the duct flow problem. However, the inlet conditions for the duct flow are somewhat different from those of the wedge problem. The inlet velocity profile for the duct flow may be parabolic. Therefore, $\tilde{f}$ must be scaled to the appropriate velocity values during the process described in Equation D.2.

$\tilde{v}$ term

The $\tilde{v}$ term is calculated from the following equation:

$$\tilde{v} = \frac{1}{\Delta t}\begin{bmatrix} \widetilde{M} & 0 & 0 \\ 0 & \widetilde{M} & 0 \\ 0 & 0 & 0 \end{bmatrix}\begin{bmatrix} u_1^n \\ u_2^n \\ 0 \end{bmatrix} = \begin{bmatrix} \frac{1}{\Delta t}\widetilde{M} & 0 & 0 \\ 0 & \frac{1}{\Delta t}\widetilde{M} & 0 \\ 0 & 0 & 0 \end{bmatrix}\begin{bmatrix} u_1^n \\ u_2^n \\ 0 \end{bmatrix} \tag{E.36}$$

where n is the iteration number.

As stated above, from the boundary conditions it is assumed that at $t = 0$ the fluid is in equilibrium. The velocity and pressure at each point is zero. It is also assumed that, for the lid-driven cavity flow cases the bottom lid starts moving with an x-axial velocity of unity ($u = 1\hat{x}$).

In reality there would be an acceleration phase for the lid before reaching this constant velocity. However, in these calculations the acceleration is neglected and instantaneous velocity for the bottom lid is assumed. For the duct flow it is also assumed that the velocity is zero at the nodal points at $t = 0$.

Assembly of the convection term

The convection term in the Navier-Stokes equations is represented in the form of:

$$-\frac{1}{12}(23C^n - 16C^{n-1} + 5C^{n-2}) \tag{E.37}$$

where $C^q(q = n,\ n-1,\ n-2)$ are obtained from direct-stiffness summation of

$$\sum_{\delta=1}^{6} \left(\hat{C}^k_{\alpha\delta}\right)^q u^q_{g^k_\delta} \tag{E.38}$$

over k and α. The C^q term is described in Section E.1.1.

Once the elemental convection vectors are assembled, the global vector is obtained from the summation logic given below. The assembly is similar to the assembly process of the global stiffness matrix.

zero $(\hat{C}U1\_GLOB)$ and $(\hat{C}U2\_GLOB)$

{For $1 \leq k \leq$ Number of elements (K)

{For $1 \leq \alpha \leq$ (Number of local velocity nodes (=6))

$\{j = (g^V)^k_\beta$

$(\tilde{C}U1\_GLOB)_j = (\tilde{C}U1\_GLOB)_j + (\hat{C}U1\_ELEM)_{k,\alpha}$

$(\tilde{C}U2\_GLOB)_j = (\tilde{C}U2\_GLOB)_j + (\hat{C}U2\_ELEM)_{k,\alpha}$

} } } (E.39)

finally the system convection term is assembled from the global terms

$$C = \begin{bmatrix} (\tilde{C}U1\_GLOB) \\ \vdots \\ (\tilde{C}U2\_GLOB) \\ \vdots \\ 0 \\ \vdots \end{bmatrix} \tag{E.40}$$

This convection term is calculated for a current and two previous time steps. During the first three steps there are not enough components to calculate the convection term. For the first time step the initial velocities are zero. Therefore, for the first time step the convection term is zero. For the second and third time steps only the last component of the convection term is used without multipliers. After the fourth iteration step there are enough previous convection term results to properly calculate the full convection term as given above.

E.2 FEM mesh generation

E.2.1 Structured mesh generation for regular domains

The method used in this thesis, to solve the Navier–Stokes problem for various geometries, such as for lid-driven triangular and rectangular domains (see Figures 3.1 and 3.2) and for duct flow cases (see Figure 3.3) require the FEM discretization of the domain. The domain is divided into structured finite triangular Taylor–Hood elements ($\mathbb{P}_2$ — $\mathbb{P}_1$ velocity–(continuous) pressure FEM implementation), which employs quadratic velocity nodes and linear pressure nodes.

For any given triangular element, quadrature represents 6 elemental nodes, one for each vertex of the triangular element, and one for each mid-point between the vertices. For triangular linear nodes, 3 elemental nodes are used, one at each vertex of the triangle. To account for these correctly and to reduce the complexity of the mesh generator, two separate mesh generators were created each, for both domain shapes. One to calculate linear pressure nodes and one to calculate quadratic velocity nodes and related information, such as node connectivity. Node

connectivity establishes the map between local and global nodes.

The main benchmark and case geometries are described in Section 1.3. They can be summarized as triangular and rectangular lid-driven cavity flows, duct flow (Poiseuille flow), backward and forward facing steps, a vertical wall in a duct, duct with a split inlet and split inlet model of the upper airway.

The mesh generators calculate the velocity node and pressure node geometries (through two separate subroutine calls). These subroutines assemble the global node matrices and the connectivity matrices for both velocity and pressure nodes. The returned global node matrices include geometric properties for each node. The information for pressure nodes are given in brackets. Note that these represent different returned variables from the ones returned for the velocity nodes. The returned properties are:

- global velocity (and pressure) node numbers
- x-coordinate of the velocity (pressure) node in the global scale
- y-coordinate of the velocity (pressure) node in the global scale
- *mask* of the node (0 for boundary nodes, 1 for interior nodes) for the lid-driven cavity flow cases
- *mskbot* of the moving lid nodes (0 for the lid boundary nodes, 1 otherwise) for the lid-driven cavity flow cases
- *maskwall* of the node (0 for wall boundary nodes, 1 for interior nodes) for the duct flow cases
- *maskinlet* of the node (0 for inlet boundary nodes, 1 otherwise) for the duct flow cases
- *maskoutlet* of the node (0 for outlet boundary nodes, 1 otherwise) for the duct flow cases

The connectivity matrix provides a mapping between the global velocity (pressure) nodes and corresponding elements and local node numbers.

Node numbering schemes for the lid-driven triangular cavity cases are shown in Figures E.3 and E.4 for velocity and pressure nodes, respectively. Here, the number of blocks in the y direction define the number of blocks in the x direction, due to the geometry of the domain. While developing the code for calculating connectivity, the domain was sub-divided into 4 distinct sections, due to the geometry of the domain and the numbering scheme used throughout the calculations. These sub-divisions were:

- boundary elements on the left side of the domain
- boundary elements on the right side of the domain
- upper triangular elements within a 2 element block in the middle of the domain
- lower triangular elements within a 2 element block in the middle of the domain

For the lid-driven rectangular cavity flow cases the number of blocks in the y direction are set equal to the number of blocks in the x direction due to the geometry of the domain. (See Figures E.5 and E.6.)

The discretization scheme for simple duct flow cases is identical to the one for the lid-driven rectangular cavity domain. The difference lies in the assignment of *mask* flags identifying the boundary conditions. For a lid-driven case the domain is surrounded by 3 walls and one moving boundary. For the duct flow case the top and bottom walls are non moving, while the inlet and outlet conditions affect the left and right sides of the domain.

For the backward and forward facing steps, the domain was divided to two sub-domains, one including the step and the other containing the rest of the domain. The nodalization was similar to the one described for the rectangular domain. However, in an additional processing

Figure E.3: Velocity node numbering scheme for a wedge

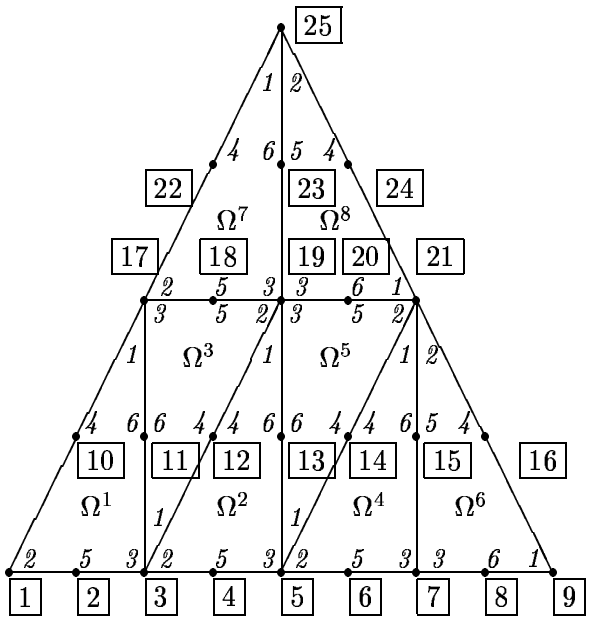


Figure E.4: Pressure node numbering scheme for a wedge

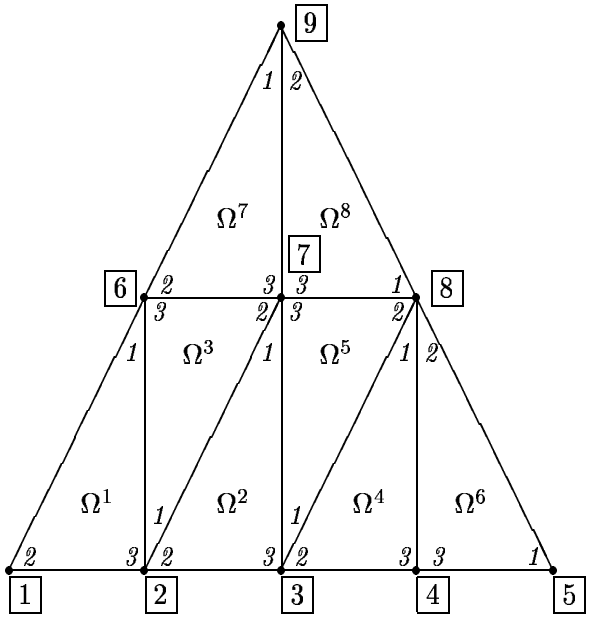


Figure E.5: Velocity node numbering scheme for a 2D duct

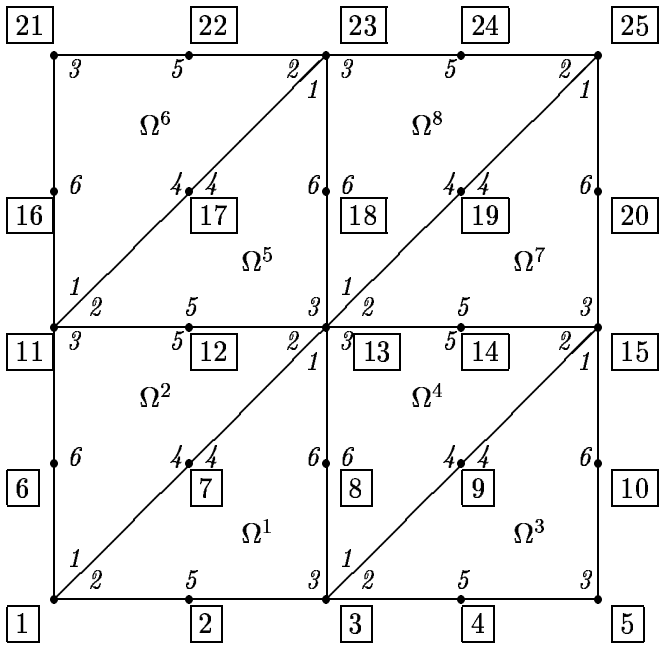
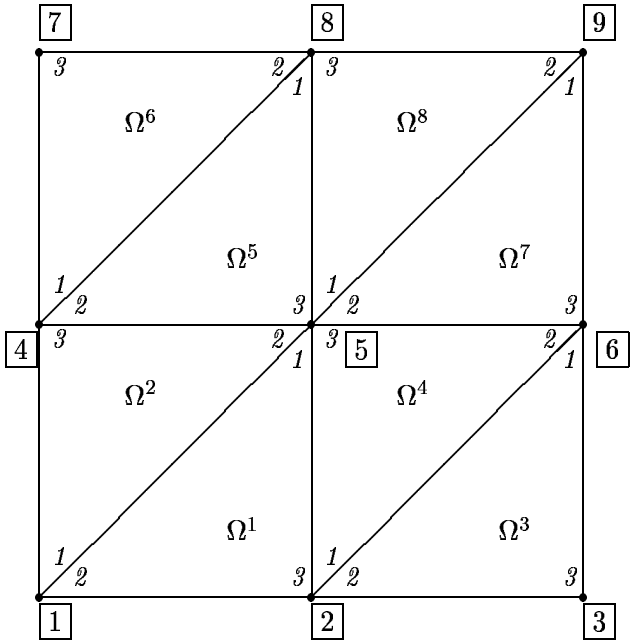


Figure E.6: Pressure node numbering scheme for a 2D duct



step, the nodes were re-numbered to form a continuous set and the two sub-domains were glued together to form a complete domain. It is similar to the process demonstrated for split domains below.

For the vertical wall and split inlet cases the domain is split, while neighboring elements seemingly share the same nodes. Such nodalization would be incorrect, because the elements must be physically separated. To achieve this, a numbering scheme was employed, which is shown in Figures E.7 and E.8. In Step 1, the domain is divided into sub-domains, and numbered. In Step 2, the sub-domains are glued together forming a continuous domain, while maintaining separation between the appropriate sub-domains.

Note that in the numerical scheme the pressure is set to 0 at one node. To avoid numerical difficulties, it is advised to develop a node numbering scheme where the last node is set to the location where the reference pressure is calculated, for example at an exit corner of the domain.

E.2.2 Structured mesh generation for irregular domains

The generic duct grid is created as a template mesh. However, the final mesh may have different domain boundary coordinates, and different shapes including obstructions. For example, modeling the upper airway results in varying cross-sectional diameters along the axis. In these cases the generic mesh and the affected nodal points are stretched to the desired geometry, using a "modifier", which keeps the global and local node and element numbering, but it changes the domain shape according to predefined equations. The changed variables are the x and y coordinates. The concept is demonstrated using a simple obstruction in a duct. Figure E.9 shows the unobstructed domain using only 260 elements for clarity. Figure E.10 shows the same domain modified to include a symmetrical wedge shape obstruction. While the number of elements are kept the same, the area changed for the affected elements. Therefore, for the dynamic cases, where the wall compliance changes the boundaries, the domain must be

Figure E.7: Step 1 of node re-numbering for a split inlet domain

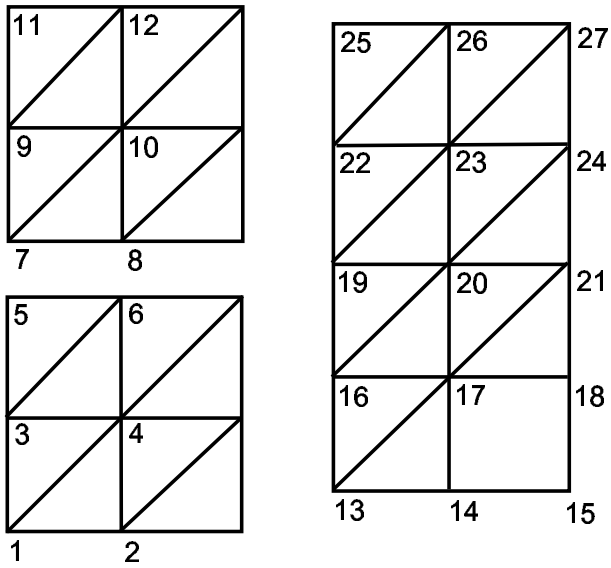


Figure E.8: Step 2 of node re-numbering for a split inlet domain

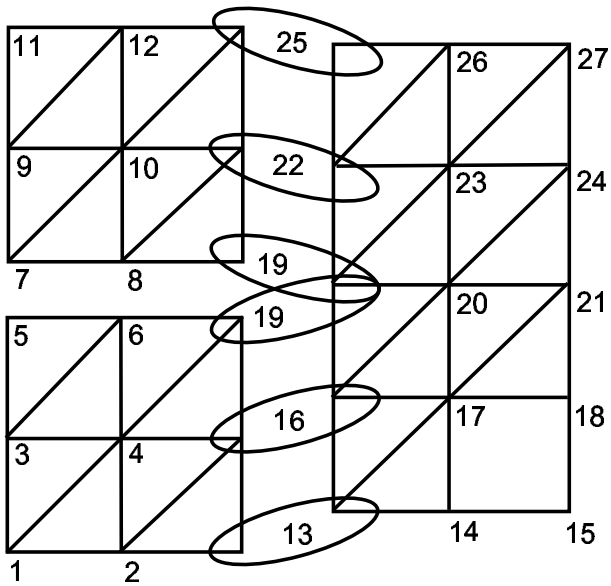
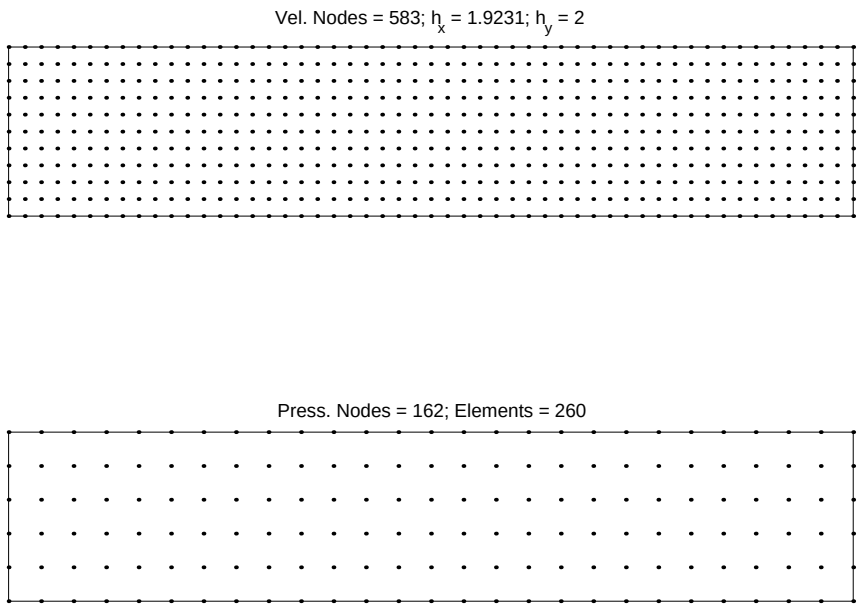


Figure E.9: Unobstructed duct domain



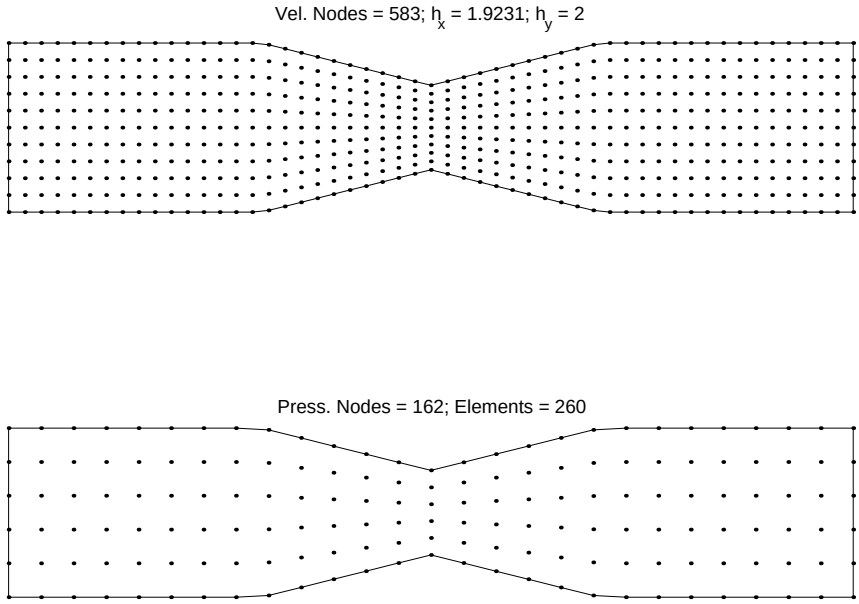
re-meshed at each time step, and the flow areas and nodal coordinates must be re-calculated. This is further explained in the next section dealing with the coupled problems.

E.2.3 Structured mesh generation for the coupled code

The mesh generator for the split-inlet domain was updated to account for the non-uniform displacement of the cantilevered plate section. (That is the end of the horizontal split.) At a given location along the x-axis, for each plate nodal point, the y-coordinates were proportionately reduced or expanded above or below the plate, based on a displacement vector array. The initial displacement of the cantilevered plate is generated outside of the code for a required eigenmode, using a displacement array generator. During time series calculations, the initial displacement array is replaced with the displacement, which is calculated in the previous time step and used to re-mesh the domain.

Examples for the modified pressure node mesh are shown in Figure E.11 for Mode 1, Mode 2 and Mode 3 initial displacements.

Figure E.10: Modified symmetrically obstructed domain



Similarly, the mesh generator was modified to account for the upper airway (vocal tract) geometries, coupled with a displaced soft palate. Examples of pressure node grids with 2560 elements, for Mode 1, Mode 2 and Mode 3 initial displacements are shown in Figure E.12. For both duct configurations the velocity node mesh looks similar, but with a higher node density. The vocal tract geometries are based on data from Reference [DHV+99].

The mesh generator for the pressure nodes also returned a two-dimensional array with the node numbers at the matching top and bottom nodes along the flexible plate. This helped to determine the pressure difference between the top and the bottom nodes of the plate. The array is used in the wall code – with the flexural rigidity and damping force – to determine the displacement of the plate at each time step.

E.2.4 Masks

There are several masks employed in the block formation process. These are the *mask*, *mskbot* and *pressure_mask* for the lid-driven cavity flow cases. In addition to these, *maskwall*,

Figure E.11: Pressure node mesh of a 2D duct – split inlet, displaced cantilevered plate

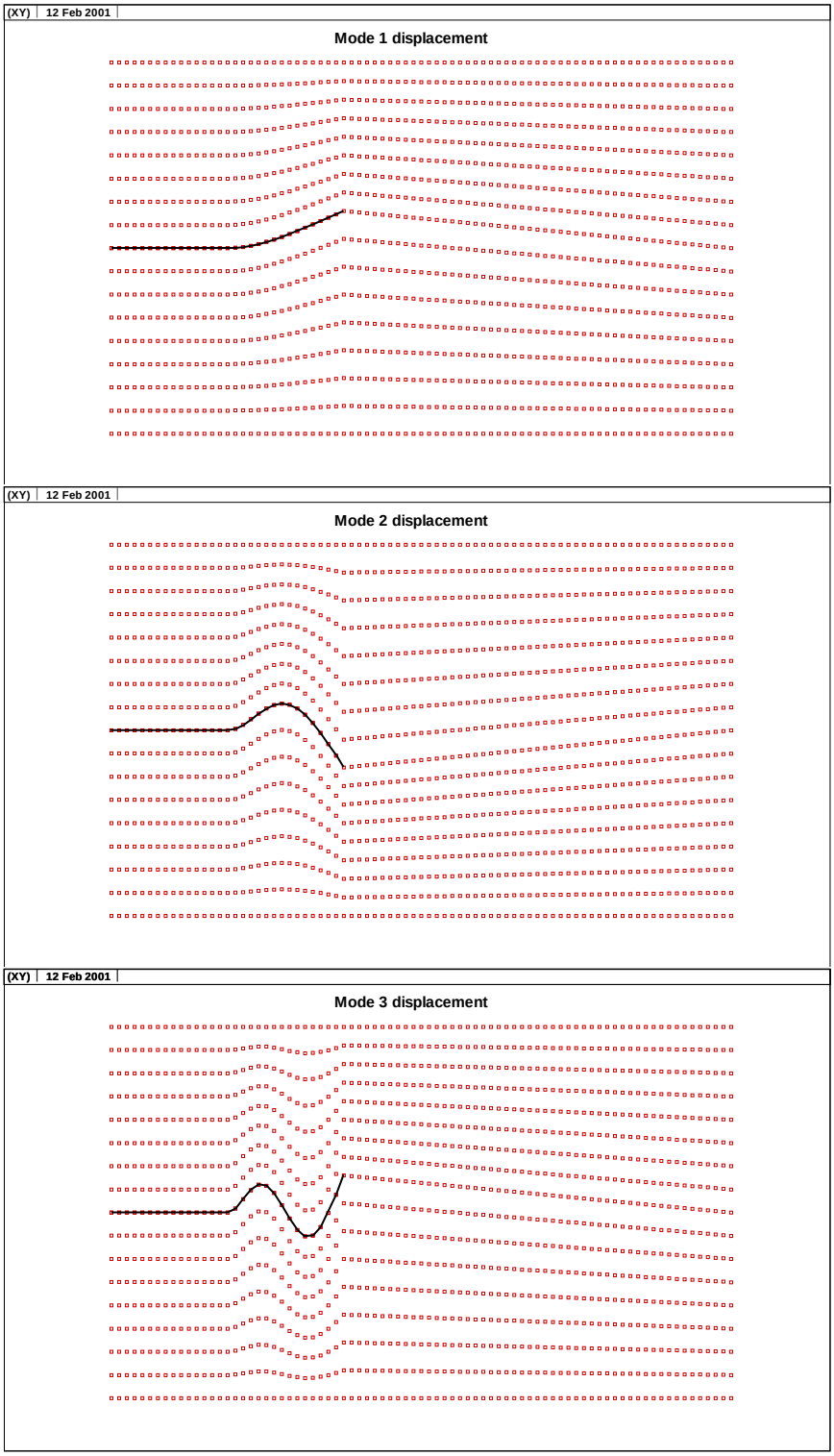
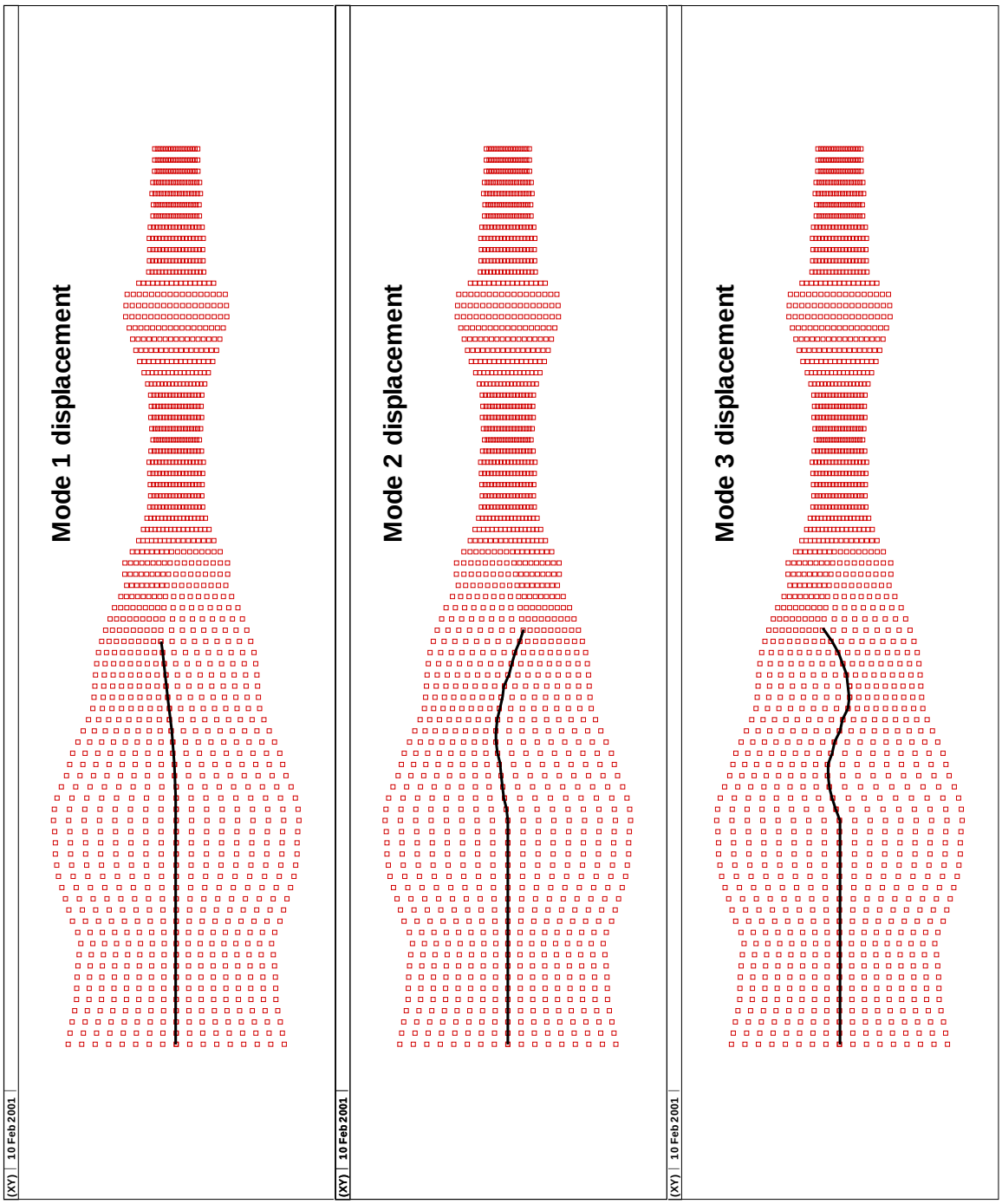


Figure E.12: Pressure node meshes for the upper airway with a displaced soft palate



maskinlet and *maskoutlet* for the duct flow cases. These masks have no mathematical meaning. Instead, they are used as "flags" to identify special nodes such as boundary nodes, and interior nodes during the block formation process.

Mask

The *mask* is used in the calculations for a lid-driven cavity flow cases. The *mask* array is defined as:

$$mask_i = \begin{cases} 0 \; x_i^V & \in \partial\Omega \\ 1 \; x_i^V & \notin \partial\Omega \end{cases} \tag{E.41}$$

meaning that the *mask* is set to 0 for the boundary velocity nodes and set to 1 for the interior nodes.

Mskbot

For the lid-driven cavity flow cases the nodes associated with the moving lid are distinguished by the *mskbot* array, that is

$$mskbot_i = \begin{cases} 0 \; x_i^V & \in \partial\Omega_{lid} \\ 1 \; x_i^V & \notin \partial\Omega_{lid} \end{cases} \tag{E.42}$$

meaning that the values are set to 0 for the moving lid nodes and set to 1 for the rest of the nodes.

Pressure mask

The pressure mask is used in the block formation process to identify (any) one pressure node. For simplicity, the last vector element of the pressure mask vector is set to zero, while all other elements are set to unity. When the system matrix is reduced, by removing the appropriate rows and columns, corresponding to the known boundary conditions, and the elements from

the forcing term, the pressure mask flags one pressure node. Consequently, the corresponding elements from the forcing term and the rows and columns from the system matrix are removed. This is required to solve the system of equations.

Wall mask

The wall mask is used for the "duct" flow cases and it is similar to the *mask* given above. The value of the wall mask is zero at the wall and unity at other nodes. Its values are given by the mesh generator for the duct domain. That is

$$maskwall_i = \begin{cases} 0 \; x_i^V & \in \partial\Omega_{wall} \\ 1 \; x_i^V & \notin \partial\Omega_{wall} \end{cases} \tag{E.43}$$

Inlet mask

The inlet mask is used for the "duct" domain to identify the "inlet" nodes. The value of the inlet mask is zero at the inlet of the duct and unity at other nodes. Its values are given by the mesh generator for the duct domain. That is

$$maskinlet_i = \begin{cases} 0 \; x_i^V & \in \partial\Omega_{inlet} \\ 1 \; x_i^V & \notin \partial\Omega_{inlet} \end{cases} \tag{E.44}$$

Outlet mask

The outlet is used for the "duct" domain to identify the "outlet" nodes. The value of the outlet mask is zero at the outlet nodes and unity at other nodes. Its values are given by the mesh generator for the duct.

$$maskoutlet_i = \begin{cases} 0 \; x_i^V & \in \partial\Omega_{outlet} \\ 1 \; x_i^V & \notin \partial\Omega_{outlet} \end{cases} \tag{E.45}$$

E.3 Solving the FEM System of Equations

The matrices resulting from the FEM formulation, using a system of equations are sparse. That is, most of the elements of the matrices are identically zero. The simplest example for this is a band matrix. Such a matrix is obtained from the present FEM formulation of the Stokes or Navier-Stokes equations.

In this, the system matrix is solved using a direct method, which is a variation of the Gaussian discretization. For practical execution of the Gauss elimination process, the following theorem is used:

`Theorem`: Each non-singular matrix A can be written as:

$$A = LU \tag{E.46}$$

where L is a lower triangular matrix with unit main diagonal and U is an upper triangular matrix. This expression is exploited in the following manner:

Let

$$A\underline{x} = \underline{f} \tag{E.47}$$

be a system of equations to be solved.

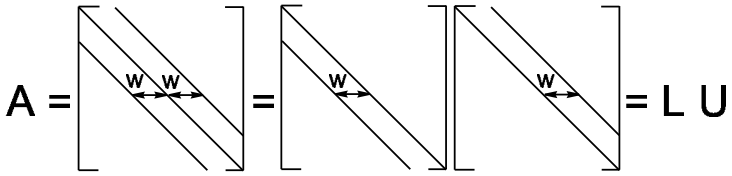
Define

$$U\underline{x} = \underline{y} \tag{E.48}$$

then

$$L\underline{y} = \underline{f} \tag{E.49}$$

Figure E.13: A band matrix and its factors, with half bandwidth w



From the first equation $\underline{y}$ can be solved directly since L is a lower triangular matrix, and subsequently $\underline{x}$ can be solved from the second equation starting from the bottom. This latter is called back-substitution. The matrices L and U are also band matrices, as shown in Figure E.13. In fact the discretization by FEMs, when a "rectangular" regular mesh is used, turns out to be exactly a band matrix. In this thesis, structured grid is used for both triangular and rectangular domains. (Note that for irregular or unstructured meshes it is useful to exploit more than the band alone. For those meshes, profile methods (or skyline methods) may be used.)

E.3.1 Direct solver

The main advantages of using direct solvers are the easy and fast programability, and the stability of the solution. It is ideal for testing new numerical schemes. However, the disadvantage is that the system matrix requires a large amount of computer memory during execution.

The general process of solving such a system of equations is outlined below. After assembling the system matrix and the right side, the resulting system to be solved is:

$$\widetilde{S} \begin{bmatrix} u_1^{n+1} \\ u_2^{n+1} \\ p^{n+1} \end{bmatrix} = [\widetilde{f} + \widetilde{v}] + \text{Conv.} \tag{E.50}$$

where u_1^{n+1} and u_2^{n+1} are the velocity vectors in the x and y directions, given at the velocity nodes, while p^{n+1} is the pressure (a scalar) at the pressure nodes (all of these at iteration step $n+1$). Conv. is the convection term given in Section E.1.2. The term $\widetilde{v}$ is calculated with

velocity values at iteration step n.

The third vector component of the forcing term which corresponds to the locations of pressure related values can be non-zero, due to the sliding lid or inlet boundary conditions. As explained above, this BC added whole columns from the system matrix to the right hand side. This most likely carried over values to the bottom of the "forcing term" as well resulting in non-zero entries there.

Before the use of the direct solver, the left and the right hand sides were prepared by the masks and the system matrix and the right hand side vector was reduced by the removal of the required rows and columns.

The procedure given here ensured that the boundary conditions and the pressure determinacy conditions are accounted for. First, the right side of the equation was assembled using the masks. This process was described in a previous section. Then the resulting $\widetilde{f}$ vector was multiplied by $mskbot_{big}$ to flag the boundary nodes. Next, the vector was multiplied by $pressure\_mask_{big}$ for the pressure determinacy. Before the system of equations was solved, the vector entries corresponding to the flagged elements, and the matrix row and column entries for the corresponding elements were removed. A direct solver with MATLAB is achieved by dividing the reduced system matrix with the forcing term in the form of:

$$\widetilde{u}^{n+1} = \widetilde{S} \setminus [\widetilde{F} + \text{Conv.}] \tag{E.51}$$

where $\widetilde{u}^{n+1}$ is the solution vector for u_1^{n+1}, u_2^{n+1} and p^{n+1}, at the $n+1$ iteration step.

In FORTRAN, two IMSL Math library functions were used. The first for the LU decomposition of the system matrix (DLFTXG) and the second, a sparse matrix solver (DLFSXG), to solve the resulting decomposed system matrix.

Finally, the results were mapped back from the reduced matrix (solved with the direct solver) to the original node numbering, and most importantly, the boundary conditions for the sliding lid or the inlet velocities were replaced in the solution vector.

E.3.2 Stopping criterion

The unsteady Stokes problem and the Navier-Stokes equations require iterative solutions. When a steady-state solution is required, a stopping criterion is used to terminate the calculations. Hence, convergence was achieved through satisfying an RMS condition. The Root Mean Square condition for the explicit scheme was considered satisfied, when the summed up RMS error – calculated between velocity values at subsequent time steps reached a given limit. The RMS condition was calculated from

$$RMS = \sqrt{\frac{\sum (u^{n+1} - u^n)^2}{\sum (u^{n+1})^2}} \leq RMS_W \tag{E.52}$$

Typical values for RMS convergence criterion were set between $RMS_W = 1 \times 10^{-5}$ and $RMS_W = 5 \times 10^{-5}$. The RMS changes were calculated for the u_1, u_2 and for the total changes. When the criterion was reached, it was assumed that convergence was reached.

It should be remembered that the finite element method minimizes the residual of the approximation of the differential equation over each element - not the error in the nodal value. In other words, the nodal values are adjusted such that the weighted-integral of the error in the approximation of the differential equation is minimized over each element.

E.3.3 Mesh Size

For the system of equations which arise in the discretization of the Navier–Stokes equations the matrix size is limited by the memory available for the computation. The sparseness of the system matrix can be utilized to reduce computational time and memory requirements during

the calculations.

The system matrix for the Navier–Stokes problem sets is given in Equation E.32. As the Navier-Stokes solver was ported from Matlab to FORTRAN, sparse matrices were employed in the new code to minimize memory requirements. The numerical simulations are performed on a system with 512 MB of physical memory. It was found that problems up to the physical memory limit executed extremely fast. For discretizations, where the system matrix exceed the physical memory limit, the program execution slowed down a great deal. Therefore, it was decided that the maximum element size will be kept at a level, where the physical memory is not exceeded. Practical considerations dictate that for a triangular domain 2450 elements (70 elements along the x-axis, 35 elements along the y-axis), while for a rectangular domain 2560 elements (80 element blocks along the x-axis, 16 element blocks along the y-axis) can be used. For the rectangular domain it gives a 5:1 channel aspect ratio, which can provide convenient partitioning along the x-axis.

From the discussion above the question arises: How sparse is the system matrix? Figures E.14 and E.15 illustrate the sparsity of the system matrix for a 200 element triangular domain for Stokes and Navier-Stokes problems, respectively. Note that the system matrix for the Navier-Stokes problem is somewhat larger due to the inclusion of the Mass Matrix. A summary of the full and sparse matrix sizes are presented in Table E.1. It can be seen that only a small fragment of the system matrix contains non-zero elements. In addition, the sparsity increases with the increase of the mesh size. For example, a 200 element triangular domain is about 4% sparse, while a 2450 element domain is only about 0.4% sparse. The increase of elements results in an increase of the number of velocity and pressure nodes, with the consequent increase of zero matrix elements. This then reduces the sparsity of the system matrix.

Another example can be given using a larger matrix size. A $35 \times 35 \times 2$ triangular domain

Figure E.14: Sparsity of a 200 element system matrix (Stokes problem)

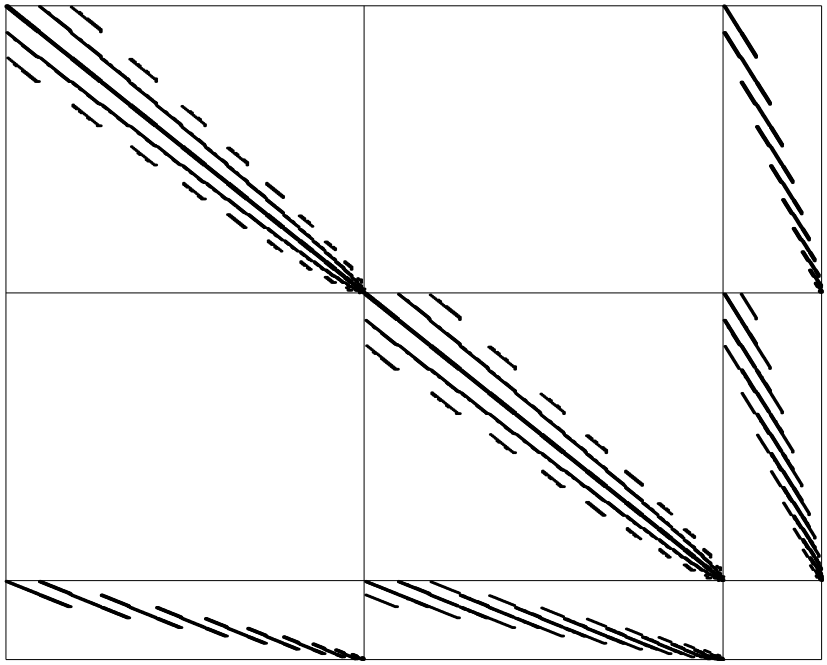


Figure E.15: Sparsity of a 200 element system matrix (Navier-Stokes problem)

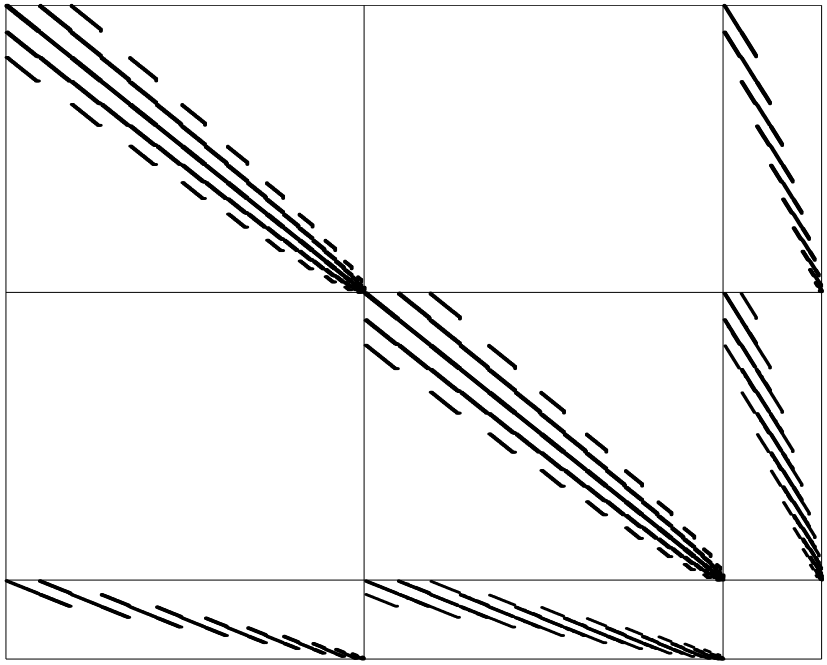


Table E.1: Triangular domain – system matrix sparsity

No. of Elements	200		1800		2450	
	Stokes	$N-S$	*Stokes*	$N-S$	*Stokes*	$N-S$
Elements along y	10	10	30	30	35	35
Velocity nodes	441	441	3721	3721	5041	5041
Pressure nodes	121	121	961	961	1296	1296
Stiffness Matrix Els.	194481	194481	13845841	13845841	25411681	25411681
Mass Matrix Els.	N/A	194481	N/A	13845841	N/A	25411681
Gradient Mat. (DD1)	53361	53361	3575881	3575881	6533136	6533136
Gradient Mat. (DD2)	53361	53361	3575881	3575881	6533136	6533136
Full Sys. Mat. Els.	1006009	1006009	70610409	70610409	124947684	124947684
Sparse Stiffness Mat.	3093	4577	25173	39417	36043	54647
Sparse Mass Matrix	N/A	3481	N/A	30841	N/A	41931
Sparse Grad.Mat. (DD1)	1056	1056	9486	9486	13034	13034
Sparse Grad.Mat. (DD2)	1243	1243	10464	10464	14116	14116
Sparse System Matrix	10784	13752	90246	118734	126386	163594
$3 *$ Sp.Sys.Mat.	32352	41256	270738	356202	379158	490782
Sparsity	3.2%	4.1%	0.38%	0.50%	0.30%	0.39%
CPU time	0.39 *sec*	0.53 *sec*	16 *sec*	31 *sec*	32 *sec*	60 sec

contains 2450 elements, which includes 5041 velocity nodes and 1296 pressure nodes. The big left hand side matrix has 129458884 matrix elements (calculated from $(2 \times 5041 + 1296)^2$) – see Equations E.32 and 3.53.

The sparse non-zero elements are calculated from the summation of the sparse Stiffness and Gradient Matrices:

$$2 * (A_{sparse} + DD1_{sparse} + DD2_{sparse}) = 2 * (54647 + 13034 + 14116) = 163594 \quad \text{(E.53)}$$

The sparse matrix is stored in 3 vector arrays, such as $[i\_row]$, $[j\_column]$, $[value\ at\ (i, j)]$.

Therefore, the required storage is $3 \times 163594 = 490782$, which is less than 0.4% of the unsparsed matrix.

It is a significant gain. Peak memory storage requirement is needed when both the global Stiffness and Mass matrices for the Navier-Stokes problems are kept in memory, yet still unsparsed. Once it is sparsed and the global Stiffness and Mass matrices are deallocated from memory, the remaining sparse vectors remain well below the peak memory usage.

It was found that problems up to about 2600 elements can be calculated efficiently within the physical memory of the computer, when systematic allocation and deallocation of the matrices are employed.

Figures E.16 and E.17 show the recommended maximum discretization of a rectangular domain (2560 elements) and a triangular domain (2450 element), respectively, using uniform meshes. Further improvement to the mesh can be achieved by local mesh refinement, that is, by reducing the mesh size closer to the wall and increasing it in the centre of the domain, while keeping the same number of elements.

Figure E.16: Max. Recommended mesh size for a rectangular domain (H=16 el.; L=80 el.)

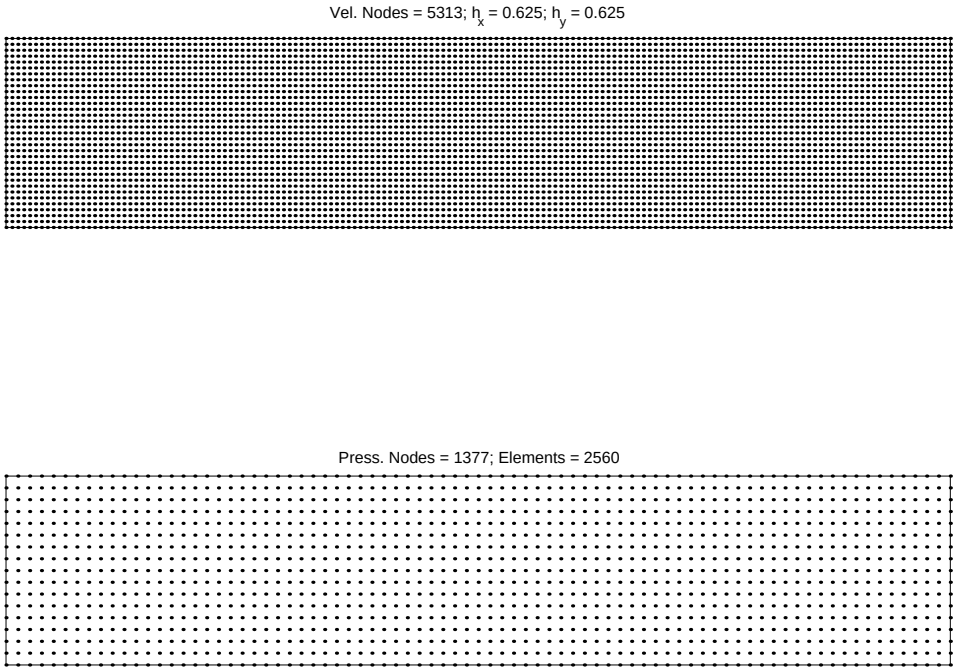
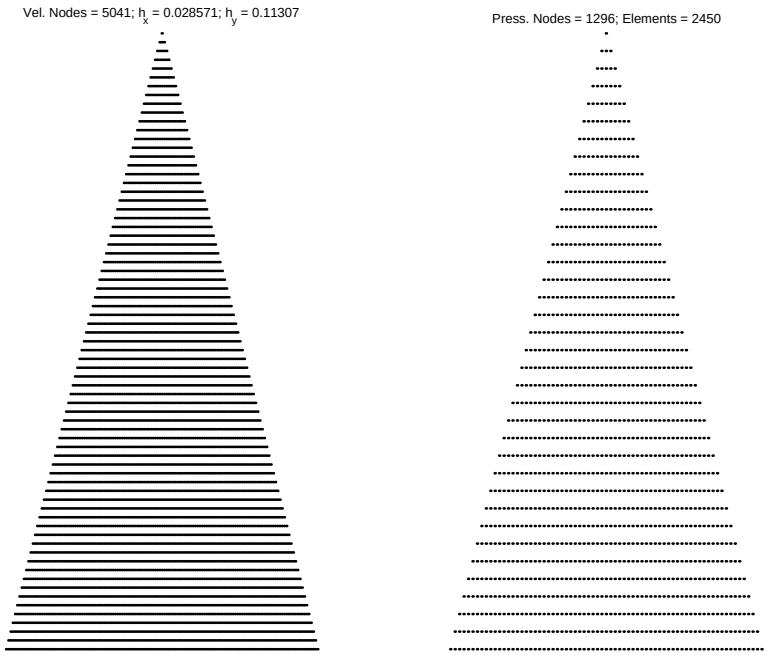


Figure E.17: Max. Recommended mesh size for a triangular domain (2450 elements)



An algorithm was developed for this study, which utilizes the capabilities of the FORTRAN 90/95 programing language standard, to minimize memory usage, by dynamically allocating and deallocating matrices. This algorithm is given in Appendix K. Such a process allowed to increase the mesh size more than ten fold, to the numbers recommended above.

E.3.4 Iterative solvers

Another way to increase the mesh size is by re-formulating the numerical scheme to use an iterative solver instead of a direct solver. For simple equations, such as the Poisson equation, a matrix free conjugate gradient method can be used. The algorithm for the conjugate gradient method is given in Appendix L.

For the Navier–Stokes problem the Uzawa method can be used, which is based on multiple loops of the conjugate gradient method. Further information on the conjugate gradient and Uzawa methods can be found in [Bra97].

Appendix F

$G_{\alpha\beta nn'}$ Term in the Elemental Stiffness Matrix

F.1 $G_{\alpha\beta nn'}$ term

The $G_{\alpha\beta nn'}$ term in the Elemental Stiffness Matrix "A" was calculated using a MAPLE script. This sub-section gives a step-by-step walk-through of the script, with additional explanations.

Note that the area multiplier (i.e., "$2*area^k$" where k is the element number) will be accounted for in the calculation of the Elemental Stiffness Matrix.

Before begin the calculations, clear all variables from memory:

```
> restart;
```

Set the array for reference elemental axes z1 and z2 (which refer to $\bar{\xi}_1$ and $\bar{\xi}_2$ or simply, the "x" and "y" coordinates in the reference elemental space), respectively

```
> z:=array(1..2);
```

$$z := \text{array}(1..2, [])$$

Set the array for the reference elemental basis. For linear triangular elements, first introduced in the lecture notes, there are 3 nodes and the array size is set from 1 to 3. However, in this problem set, for the velocity nodes (which are used in the assembly of the Stiffness Matrix) quadratic triangular elements are used, with nodes between the corner elements. Therefore,

for quadratic triangular elements, the array size for the reference elemental basis is 6, due to the 6 nodes corresponding to each element.

```
> h:=array(1..6);
```

$$h := \text{array}(1..6,\ [])$$

The G term of the Stiffness Matrix is given in the problem description in the following form:

$$G_{\alpha\beta nn'} = \int_0^1 d\xi_2 \int_0^{1-\xi_2} \frac{\partial h_\alpha}{\partial \xi_n} \frac{\partial h_\beta}{\partial \xi_{n'}} d\xi_1$$

The G term in the Stiffness Matrix is a multi-dimensional array with the following dimensions, where α and β varies from 1 to 6, and n and n' varies between 1 and 2.

```
> g:=array(1..6,1..6,1..2,1..2);
```

$$g := \text{array}(1..6,\ 1..6,\ 1..2,\ 1..2,\ [])$$

The reference elemental basis from Element 1 to 6 are expressed below. Verification of these correlations are given in Appendix G.

```
> h[1]:=z[1]*(-1+2*z[1]);
```

$$h_1 := z_1\,(-1 + 2\,z_1)$$

```
> h[2]:=z[2]*(-1+2*z[2]);
```

$$h_2 := z_2\,(-1 + 2\,z_2)$$

```
> h[3]:=(1-z[1]-z[2])*(-1+2*(1-z[1]-z[2]));
```

$$h_3 := (1 - z_1 - z_2)\,(1 - 2\,z_1 - 2\,z_2)$$

```
> h[4]:=4*z[1]*z[2];
```

$$h_4 := 4\,z_1\,z_2$$

```
> h[5]:=4*z[2]*(1-z[1]-z[2]);
```

$$h_5 := 4\,z_2\,(1 - z_1 - z_2)$$

```
> h[6]:=4*z[1]*(1-z[1]-z[2]);
```

$$h_6 := 4\,z_1\,(1 - z_1 - z_2)$$

Set up loops for α, β, n1 (or n) and n2 (or n'), and calculate G at these array elements:

```
> for alpha from 1 by 1 to 6 do:

> for beta from 1 by 1 to 6 do:

> for n1 from 1 by 1 to 2 do:

> for n2 from 1 by 1 to 2 do:

> eq1:=diff(h[alpha],z[n1]);

> eq2:=diff(h[beta],z[n2]);

> eq3:=eq1*eq2;eq4:=int(eq3,z[1]=0..1-z[2]);

> g[alpha,beta,n1,n2]:=int(eq4,z[2]=0..1);

> unassign('z');z:=array(1..2);

> od;od;od;od;
```

Read in the libraries for "C" for conversion of the outputs to "C" language, and save the values for G in both "C" and "FORTRAN" languages.

```
> readlib(C):

> C(g,filename="I:/Thesis/Courses/CFM/Problem2/P2b_support/gfile.c");
> fortran(g,filename="I:/Thesis/Courses/CFM/Problem2/P2b_support/gfile.
> f");
```

F.2 Output Variables for $G_{\alpha\beta nn'}$ (FORTRAN version)

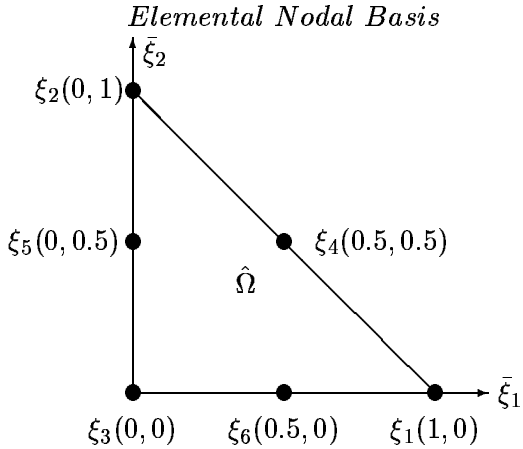
$g(1,1,1,1) = 1.E0/2.E0$
$g(1,1,1,2) = 0$
$g(1,1,2,1) = 0$
$g(1,1,2,2) = 0$
$g(1,2,1,1) = 0$
$g(1,2,1,2) = -1.E0/6.E0$
$g(1,2,2,1) = 0$
$g(1,2,2,2) = 0$
$g(1,3,1,1) = 1.E0/6.E0$
$g(1,3,1,2) = 1.E0/6.E0$
$g(1,3,2,1) = 0$
$g(1,3,2,2) = 0$
$g(1,4,1,1) = 0$
$g(1,4,1,2) = 2.E0/3.E0$
$g(1,4,2,1) = 0$
$g(1,4,2,2) = 0$
$g(1,5,1,1) = 0$
$g(1,5,1,2) = 0$
$g(1,5,2,1) = 0$
$g(1,5,2,2) = 0$
$g(1,6,1,1) = -2.E0/3.E0$
$g(1,6,1,2) = -2.E0/3.E0$
$g(1,6,2,1) = 0$
$g(1,6,2,2) = 0$
$g(2,1,1,1) = 0$
$g(2,1,1,2) = 0$
$g(2,1,2,1) = -1.E0/6.E0$
$g(2,1,2,2) = 0$
$g(2,2,1,1) = 0$
$g(2,2,1,2) = 0$
$g(2,2,2,1) = 0$
$g(2,2,2,2) = 1.E0/2.E0$
$g(2,3,1,1) = 0$
$g(2,3,1,2) = 0$
$g(2,3,2,1) = 1.E0/6.E0$
$g(2,3,2,2) = 1.E0/6.E0$
$g(2,4,1,1) = 0$
$g(2,4,1,2) = 0$
$g(2,4,2,1) = 2.E0/3.E0$
$g(2,4,2,2) = 0$
$g(2,5,1,1) = 0$
$g(2,5,1,2) = 0$
$g(2,5,2,1) = -2.E0/3.E0$
$g(2,5,2,2) = -2.E0/3.E0$
$g(2,6,1,1) = 0$
$g(2,6,1,2) = 0$
$g(2,6,2,1) = 0$
$g(2,6,2,2) = 0$
$g(3,1,1,1) = 1.E0/6.E0$
$g(3,1,1,2) = 0$
$g(3,1,2,1) = 1.E0/6.E0$
$g(3,1,2,2) = 0$
$g(3,2,1,1) = 0$
$g(3,2,1,2) = 1.E0/6.E0$
$g(3,2,2,1) = 0$
$g(3,2,2,2) = 1.E0/6.E0$
$g(3,3,1,1) = 1.E0/2.E0$
$g(3,3,1,2) = 1.E0/2.E0$
$g(3,3,2,1) = 1.E0/2.E0$
$g(3,3,2,2) = 1.E0/2.E0$
$g(3,4,1,1) = 0$
$g(3,4,1,2) = 0$
$g(3,4,2,1) = 0$
$g(3,4,2,2) = 0$
$g(3,5,1,1) = 0$
$g(3,5,1,2) = -2.E0/3.E0$
$g(3,5,2,1) = 0$
$g(3,5,2,2) = -2.E0/3.E0$
$g(3,6,1,1) = -2.E0/3.E0$
$g(3,6,1,2) = 0$
$g(3,6,2,1) = -2.E0/3.E0$
$g(3,6,2,2) = 0$
$g(4,1,1,1) = 0$
$g(4,1,1,2) = 0$
$g(4,1,2,1) = 2.E0/3.E0$
$g(4,1,2,2) = 0$
$g(4,2,1,1) = 0$
$g(4,2,1,2) = 2.E0/3.E0$
$g(4,2,2,1) = 0$
$g(4,2,2,2) = 0$
$g(4,3,1,1) = 0$
$g(4,3,1,2) = 0$
$g(4,3,2,1) = 0$
$g(4,3,2,2) = 0$
$g(4,4,1,1) = 4.E0/3.E0$
$g(4,4,1,2) = 2.E0/3.E0$
$g(4,4,2,1) = 2.E0/3.E0$
$g(4,4,2,2) = 4.E0/3.E0$
$g(4,5,1,1) = -4.E0/3.E0$
$g(4,5,1,2) = -2.E0/3.E0$
$g(4,5,2,1) = -2.E0/3.E0$
$g(4,5,2,2) = 0$
$g(4,6,1,1) = 0$
$g(4,6,1,2) = -2.E0/3.E0$
$g(4,6,2,1) = -2.E0/3.E0$
$g(4,6,2,2) = -4.E0/3.E0$
$g(5,1,1,1) = 0$
$g(5,1,1,2) = 0$
$g(5,1,2,1) = 0$
$g(5,1,2,2) = 0$
$g(5,2,1,1) = 0$
$g(5,2,1,2) = -2.E0/3.E0$
$g(5,2,2,1) = 0$
$g(5,2,2,2) = -2.E0/3.E0$
$g(5,3,1,1) = 0$
$g(5,3,1,2) = 0$
$g(5,3,2,1) = -2.E0/3.E0$
$g(5,3,2,2) = -2.E0/3.E0$
$g(5,4,1,1) = -4.E0/3.E0$
$g(5,4,1,2) = -2.E0/3.E0$
$g(5,4,2,1) = -2.E0/3.E0$
$g(5,4,2,2) = 0$
$g(5,5,1,1) = 4.E0/3.E0$
$g(5,5,1,2) = 2.E0/3.E0$
$g(5,5,2,1) = 2.E0/3.E0$
$g(5,5,2,2) = 4.E0/3.E0$
$g(5,6,1,1) = 0$
$g(5,6,1,2) = 2.E0/3.E0$
$g(5,6,2,1) = 2.E0/3.E0$
$g(5,6,2,2) = 0$
$g(6,1,1,1) = -2.E0/3.E0$
$g(6,1,1,2) = 0$
$g(6,1,2,1) = -2.E0/3.E0$
$g(6,1,2,2) = 0$
$g(6,2,1,1) = 0$
$g(6,2,1,2) = 0$
$g(6,2,2,1) = 0$
$g(6,2,2,2) = 0$
$g(6,3,1,1) = -2.E0/3.E0$
$g(6,3,1,2) = -2.E0/3.E0$
$g(6,3,2,1) = 0$
$g(6,3,2,2) = 0$
$g(6,4,1,1) = 0$
$g(6,4,1,2) = -2.E0/3.E0$
$g(6,4,2,1) = -2.E0/3.E0$
$g(6,4,2,2) = -4.E0/3.E0$
$g(6,5,1,1) = 0$
$g(6,5,1,2) = 2.E0/3.E0$
$g(6,5,2,1) = 2.E0/3.E0$
$g(6,5,2,2) = 0$
$g(6,6,1,1) = 4.E0/3.E0$
$g(6,6,1,2) = 2.E0/3.E0$
$g(6,6,2,1) = 2.E0/3.E0$
$g(6,6,2,2) = 4.E0/3.E0$

Appendix G

Verification of Quadratic Reference Elemental Basis

This appendix provides the verification of the reference elemental basis, i.e., $h_i, i = 1, ..., 6$.

We begin by introducing a six node element, with elemental nodes at vertices and mid-sides, (in the figure the nodes are already mapped into the elemental nodal basis, and the nodes are shown with their respective nodal coordinates in two dimensions).



Then, we introduce the reference element space X_h^{el}

$$I\!P_2(\hat{\Omega}) = \{\hat{v} \in I\!P_2(\Omega^k)\} \qquad (i.e., \hat{v} = \hat{a} + \hat{b}\bar{\xi}_1 + \hat{c}\bar{\xi}_2 + \hat{d}\bar{\xi}_1\bar{\xi}_2 + \hat{e}\bar{\xi}_1^2 + \hat{f}\bar{\xi}_2^2) \tag{G.1}$$

Note that in Equation (B.1) the subscripts for $\bar{\xi}_1$ and $\bar{\xi}_2$ refer to the " x " and " y " coordinates, respectively, while in the figure the subscipts for ξ_1 to ξ_6 refer to the element numbers.

G.1 Verification Results

The calculations were performed using the symbolic mathematics program MAPLE V 5.1. This documentation follows the input commands and outputs from the MAPLE script "*P2a_maple.mws*".

The calculations begin with clearing all variables from memory:

```
> restart;
```

and loading the linear algebra package.

```
> with(linalg):
Warning, new definition for norm
Warning, new definition for trace
```

G.1.1 Matrix for the System of algebraic equations

From the node coordinates of the elemental nodal basis (See Figure above), substituted into Equation B.1, a system of algebraic equations is assembled in the form of a matrix.

```
> A:=matrix([[1.0,1.0,0.0,0.0,1.0,0.0],[1.0,0.0,1.0,0.0,0.0,1.0],[1.0,0
> .0,0.0,0.0,0.0,0.0],[1.0,0.5,0.5,0.25,0.25,0.25],[1.0,0.0,0.5,0.0,0.0,
> 0.25],[1.0,0.5,0.0,0.0,0.25,0.0]]);
```

$$A := \begin{bmatrix} 1.0 & 1.0 & 0 & 0 & 1.0 & 0 \\ 1.0 & 0 & 1.0 & 0 & 0 & 1.0 \\ 1.0 & 0 & 0 & 0 & 0 & 0 \\ 1.0 & .5 & .5 & .25 & .25 & .25 \\ 1.0 & 0 & .5 & 0 & 0 & .25 \\ 1.0 & .5 & 0 & 0 & .25 & 0 \end{bmatrix}$$

G.1.2 Node 1

The equations are solved for each nodal point. While matrix A remains unchanged, the forcing term on the right hand side of the equations changes from node to node. For a given node, the vector element associated with the set of equations is set to unity, while the rest of the vector elements are set to zero. For example, for the FIRST NODE, the forcing term vector can be expressed as:

```
> f1:=vector([1.0,0,0,0,0,0]);
```

$$f1 := [1.0,\ 0,\ 0,\ 0,\ 0,\ 0]$$

Once the two sides of the linear system of equations are defined, the equations are solved with MAPLE's linear equations solver function:

```
> c1:=linsolve(A,f1);
```

$$c1 := [0,\ -1.000000000,\ 0,\ 0,\ 2.000000000,\ 0]$$

From this the reference elemental basis for Node 1 can be expressed as:

```
> h1:=c1[1]+c1[2]*xi1+c1[3]*xi2+c1[4]*xi1*xi2+c1[5]*xi1^2+c1[6]*xi2^2;
```

$$h1 := -1.000000000\,\xi 1 + 2.000000000\,\xi 1^2$$

```
> simplify(%);
```

$$-1.\,\xi 1 + 2.\,\xi 1^2$$

G.1.3 Node 2

For the SECOND NODE, the forcing term vector can be expressed as:

```
> f2:=vector([0,1.0,0,0,0,0]);
```

$$f2 := [0,\ 1.0,\ 0,\ 0,\ 0,\ 0]$$

The correlation constants for h2 are::

```
> c2:=linsolve(A,f2);
```

$$c2 := [0,\ 0,\ -1.000000000,\ 0,\ 0,\ 2.000000000]$$

From this the reference elemental basis for Node 2 can be expressed as:

```
> h2:=c2[1]+c2[2]*xi1+c2[3]*xi2+c2[4]*xi1*xi2+c2[5]*xi1^2+c2[6]*xi2^2;
```

$$h2 := -1.000000000\,\xi 2 + 2.000000000\,\xi 2^2$$

```
> simplify(%);
```

$$-1.\,\xi 2 + 2.\,\xi 2^2$$

G.1.4 Node 3

For the THIRD NODE, the forcing term vector can be expressed as:

```
> f3:=vector([0,0,1.0,0,0,0]);
```

$$f3 := [0,\ 0,\ 1.0,\ 0,\ 0,\ 0]$$

The correlation constants for h3 are::

```
> c3:=linsolve(A,f3);
```

$$c3 := [1.000000000,\ -3.000000000,\ -3.000000000,\ 4.000000000,\ 2.000000000,\ 2.000000000]$$

From this the reference elemental basis for Node 3 can be expressed as:

```
> h3:=c3[1]+c3[2]*xi1+c3[3]*xi2+c3[4]*xi1*xi2+c3[5]*xi1^2+c3[6]*xi2^2;
```

$$h3 := 1.000000000 - 3.000000000\,\xi 1 - 3.000000000\,\xi 2 + 4.000000000\,\xi 1\,\xi 2 + 2.000000000\,\xi 1^2 + 2.000000000\,\xi 2^2$$

```
> simplify(%);
```

$$1. - 3.\,\xi 1 - 3.\,\xi 2 + 4.\,\xi 1\,\xi 2 + 2.\,\xi 1^2 + 2.\,\xi 2^2$$

G.1.5 Node 4

For the FOURTH NODE, the forcing term vector can be expressed as:

```
> f4:=vector([0,0,0,1.0,0,0]);
```

$$f4 := [0,\ 0,\ 0,\ 1.0,\ 0,\ 0]$$

The correlation constants for h4 are::

```
> c4:=linsolve(A,f4);
```

$$c4 := [0,\ 0,\ 0,\ 4.000000000,\ 0,\ 0]$$

From this the reference elemental basis for Node 4 can be expressed as:

```
> h4:=c4[1]+c4[2]*xi1+c4[3]*xi2+c4[4]*xi1*xi2+c4[5]*xi1^2+c4[6]*xi2^2;
```

$$h4 := 4.000000000\,\xi 1\,\xi 2$$

```
> simplify(%);
```

$$4.\,\xi 1\,\xi 2$$

G.1.6 Node 5

For the FIFTH NODE, the forcing term vector can be expressed as:

```
> f5:=vector([0,0,0,0,1.0,0]);
```

$$f5 := [0,\ 0,\ 0,\ 0,\ 1.0,\ 0]$$

The correlation constants for h5 are::

```
> c5:=linsolve(A,f5);
```

$$c5 := [0,\ 0,\ 4.000000000,\ -4.000000000,\ 0,\ -4.000000000]$$

From this the reference elemental basis for Node 5 can be expressed as:

```
> h5:=c5[1]+c5[2]*xi1+c5[3]*xi2+c5[4]*xi1*xi2+c5[5]*xi1^2+c5[6]*xi2^2;
```

$$h5 := 4.000000000\,\xi 2 - 4.000000000\,\xi 1\,\xi 2 - 4.000000000\,\xi 2^2$$

```
> simplify(%);
```

$$4.\,\xi 2 - 4.\,\xi 1\,\xi 2 - 4.\,\xi 2^2$$

G.1.7 Node 6

For the SIXTH NODE, the forcing term vector can be expressed as:

```
> f6:=vector([0,0,0,0,0,1.0]);
```

$$f6 := [0,\ 0,\ 0,\ 0,\ 0,\ 1.0]$$

The correlation constants for h6 are::

```
> c6:=linsolve(A,f6);
```

$$c6 := [0,\ 4.000000000,\ 0,\ -4.000000000,\ -4.000000000,\ 0]$$

From this the reference elemental basis for Node 6 can be expressed as:

```
> h6:=c6[1]+c6[2]*xi1+c6[3]*xi2+c6[4]*xi1*xi2+c6[5]*xi1^2+c6[6]*xi2^2;
```

$$h6 := 4.000000000\,\xi 1 - 4.000000000\,\xi 1\,\xi 2 - 4.000000000\,\xi 1^2$$

```
> simplify(%);
```

$$4.\,\xi 1 - 4.\,\xi 1\,\xi 2 - 4.\,\xi 1^2$$

G.2 Summary

The verification results for h1 to h6 are summarized below, respectively).

```
> simplify(h1);
```

$$-1.\,\xi 1 + 2.\,\xi 1^2$$

```
> simplify(h2);
```

$$-1.\xi 2 + 2.\xi 2^2$$

```
> simplify(h3);
```

$$1. - 3.\xi 1 - 3.\xi 2 + 4.\xi 1\,\xi 2 + 2.\xi 1^2 + 2.\xi 2^2$$

```
> simplify(h4);
```

$$4.\xi 1\,\xi 2$$

```
> simplify(h5);
```

$$4.\xi 2 - 4.\xi 1\,\xi 2 - 4.\xi 2^2$$

```
> simplify(h6);
```

$$4.\xi 1 - 4.\xi 1\,\xi 2 - 4.\xi 1^2$$

Appendix H

Derivation of the Elemental Mass Matrix using MAPLE

H.1 Mass Matrix for Quadratic Triangular Elements

This MAPLE script calculates the Mass Matrix for Quadratic Triangular Elements. The calculations are performed using the symbolic software MAPLE V 5.1.

The Mass Matrix is defined in the lecture notes as:

$$\begin{aligned} \hat{M}^k_{\alpha\beta} &= \int_{\Omega^k} h_\alpha h_\beta dxdy \\ &= 2\ area^k \int_0^1 \int_0^{1-\xi_2} h_\alpha h_\beta d\xi_1 d\xi_2 \end{aligned}$$

where α and β varies from 1 to 6, and k represents the element number. (Note that the portion of the equation following "2 $area^k$" will be referred as "m", while for the Mass Matrix "MM" will be used.)

First all variables are cleared from the system memory.

```
> restart;
```

Set the array for reference elemental axes z1 and z2 (which refer to $\bar{\xi}_1$ and $\bar{\xi}_2$ or simply, the "x" and "y" coordinates in the reference elemental space), respectively

```
> z:=array(1..2);
```

$$z := \text{array}(1..2,\ [])$$

Set the array for the reference elemental basis. For linear triangular elements there are 3 nodes, and the array size is set from 1 to 3. However, quadratic triangular elements have nodes between the corner elements. Therefore, for quadratic triangular elements, the array size for the reference elemental basis is 6, due to the 6 nodes corresponding to each element.

```
> h:=array(1..6);
```

$$h := \text{array}(1..6,\ [])$$

```
> m:=array(1..6,1..6);
```

$$m := \text{array}(1..6,\ 1..6,\ [])$$

```
> h[1]:=z[1]*(-1+2*z[1]);
```

$$h_1 := z_1\,(-1 + 2\,z_1)$$

```
> h[2]:=z[2]*(-1+2*z[2]);
```

$$h_2 := z_2\,(-1 + 2\,z_2)$$

```
> h[3]:=(1-z[1]-z[2])*(1-2*z[1]-2*z[2]);
```

$$h_3 := (1 - z_1 - z_2)\,(1 - 2\,z_1 - 2\,z_2)$$

```
> h[4]:=4*z[1]*z[2];
```

$$h_4 := 4\,z_1\,z_2$$

```
> h[5]:=4*z[2]*(1-z[1]-z[2]);
```

$$h_5 := 4\,z_2\,(1 - z_1 - z_2)$$

```
> h[6]:=4*z[1]*(1-z[1]-z[2]);
```

$$h_6 := 4\,z_1\,(1 - z_1 - z_2)$$

Set up loops for α and β and calculate m at these array elements:

```
> for alpha from 1 by 1 to 6 do;
> for beta from 1 by 1 to 6 do;
```

```
> eq1:=h[alpha]*h[beta];

> eq2:=int(eq1,z[1]=0..1-z[2]);

> m[alpha,beta]:=int(eq2,z[2]=0..1);

> unassign('z');z:=array(1..2);

> od;od;
```

Show the results for m, calculated above:

```
> m();
```

$$\begin{bmatrix} \frac{1}{60} & \frac{-1}{360} & \frac{-1}{360} & 0 & \frac{-1}{90} & 0 \\ \frac{-1}{360} & \frac{1}{60} & \frac{-1}{360} & 0 & 0 & \frac{-1}{90} \\ \frac{-1}{360} & \frac{-1}{360} & \frac{1}{60} & \frac{-1}{90} & 0 & 0 \\ 0 & 0 & \frac{-1}{90} & \frac{4}{45} & \frac{2}{45} & \frac{2}{45} \\ \frac{-1}{90} & 0 & 0 & \frac{2}{45} & \frac{4}{45} & \frac{2}{45} \\ 0 & \frac{-1}{90} & 0 & \frac{2}{45} & \frac{2}{45} & \frac{4}{45} \end{bmatrix}$$

The Mass Matrix for quadratic triangular elements is the above matrix (m) multiplied by $2 * area^k$, where k refers to an element

```
> MM:=2*area[k]*m();
```

$$MM := 2\, area_k \begin{bmatrix} \frac{1}{60} & \frac{-1}{360} & \frac{-1}{360} & 0 & \frac{-1}{90} & 0 \\ \frac{-1}{360} & \frac{1}{60} & \frac{-1}{360} & 0 & 0 & \frac{-1}{90} \\ \frac{-1}{360} & \frac{-1}{360} & \frac{1}{60} & \frac{-1}{90} & 0 & 0 \\ 0 & 0 & \frac{-1}{90} & \frac{4}{45} & \frac{2}{45} & \frac{2}{45} \\ \frac{-1}{90} & 0 & 0 & \frac{2}{45} & \frac{4}{45} & \frac{2}{45} \\ 0 & \frac{-1}{90} & 0 & \frac{2}{45} & \frac{2}{45} & \frac{4}{45} \end{bmatrix}$$

Save to results in C and FORTRAN language formats:

```
> readlib(C):
```

```
> C(m,filename="I:/Thesis/Courses/CFM/Problem2/P2b_support/mfile.c");
> fortran(m,filename="I:/Thesis/Courses/CFM/Problem2/P2b_support/mfile.
> f");
```

H.2 Mass Matrix for Linear Triangular Elements

The calculations are performed using the symbolic software MAPLE V 5.1.

The Mass Matrix is defined in the lecture notes as:

$$\begin{aligned} \hat{M}^k_{\alpha\beta} &= \int_{\Omega^k} h_\alpha h_\beta dxdy \\ &= 2\ area^k \int_0^1 \int_0^{1-\xi_2} h_\alpha h_\beta d\xi_1 d\xi_2 \end{aligned}$$

where α and β varies from 1 to 3, and k represents the element number. (Note that the portion of the equation following "2 $area^k$" will be referred as "m", while for the Mass Matrix "MM" will be used.)

First all variables are cleared from the system memory.

```
> restart;
```

Set the array for reference elemental axes z1 and z2 (which refer to $\bar{\xi}_1$ and $\bar{\xi}_2$ or simply, the "x" and "y" coordinates in the reference elemental space), respectively

```
> z:=array(1..2);
```

$$z := \text{array}(1..2,\ [])$$

Set the array for the reference elemental basis. For linear triangular elements there are 3 nodes, and the array size is set from 1 to 3.

```
> h:=array(1..3);
```

$$h := \text{array}(1..3,\ [])$$

```
> h[1]:=z[1];
```

$$h_1 := z_1$$

```
> h[2]:=z[2];
```

$$h_2 := z_2$$

```
> h[3]:=1-z[1]-z[2];
```

$$h_3 := 1 - z_1 - z_2$$

```
> m:=array(1..3,1..3);
```

$$m := \text{array}(1..3,\ 1..3,\ [])$$

Set up loops for α and β and calculate m at these array elements:

```
> for alpha from 1 by 1 to 3 do;
> for beta from 1 by 1 to 3 do;
> eq1:=h[alpha]*h[beta];
> eq2:=int(eq1,z[1]=0..1-z[2]);
> m[alpha,beta]:=int(eq2,z[2]=0..1);
> unassign('z');z:=array(1..3);
> od;od;
```

Show the results for m, calculated above:

```
> m();
```

$$\begin{bmatrix} \frac{1}{12} & \frac{1}{24} & \frac{1}{24} \\ \frac{1}{24} & \frac{1}{12} & \frac{1}{24} \\ \frac{1}{24} & \frac{1}{24} & \frac{1}{12} \end{bmatrix}$$

The Mass Matrix for linear triangular elements is the above matrix (m) multiplied by 2*area_k, where k refers to an element

```
> MM:=2*area[k]*m();
```

$$MM := 2\,area_k \begin{bmatrix} \frac{1}{12} & \frac{1}{24} & \frac{1}{24} \\ \frac{1}{24} & \frac{1}{12} & \frac{1}{24} \\ \frac{1}{24} & \frac{1}{24} & \frac{1}{12} \end{bmatrix}$$

These results are the same as the ones obtained during the lecture.

Save to results in C and FORTRAN language formats:

```
> readlib(C):

> C(m,filename="I:/Thesis/Courses/CFM/Problem2/P2b_support/m1file.c");
> fortran(m,filename="I:/Thesis/Courses/CFM/Problem2/P2b_support/m1file
> .f");
```

Appendix I

Derivation of the Elemental Gradient Matrix using MAPLE

I.1 MAPLE script to calculate $G^{\widehat{D}}_{\alpha\beta n}$ in $\widehat{D}^k_{\alpha\beta}$

This section provides a step-by-step explanation of the MAPLE script used to calculate $G^{\widehat{D}}_{\alpha\beta n}$ term, which is used in calculating $\widehat{D}^k_{\alpha\beta}$ for the Stokes and Navier–Stokes problems.
Note that the area multiplier (i.e., $2 * area^k$ where k is the element number) will be accounted for in $\widehat{D}^k_{\alpha\beta}$

Clear the variables from memory:

```
> restart;
```

Set the array for reference elemental axes z1 and z2, which refer to ξ_1 and ξ_2, respectively

```
> z:=array(1..2);
```

$$z := \text{array}(1..2, [])$$

Set the array for the reference elemental basis (for velocity and pressure nodes). For linear triangular elements there are 3 nodes, and the array size is set from 1 to 3. These are used for the pressure nodes and denoted as "hp". For quadratic triangular elements, the array size for the reference elemental basis is 6, due to the 6 nodes for the element. These are used for the velocity nodes and denoted as "hv".

```
> hv:=array(1..6);
```

$$hv := \text{array}(1..6,\ [])$$

```
> hp:=array(1..3);
```

$$hp := \text{array}(1..3,\ [])$$

The $G^{\widehat{D}}_{\alpha\beta n}$ term is a multi-dimensional array with the following dimensions:

```
> gd:=array(1..3,1..6,1..2);
```

$$gd := \text{array}(1..3,\ 1..6,\ 1..2,\ [])$$

For the **velocity nodes**: the reference elemental basis from (reference) Element 1 to 6 are expressed below. Verification of these correlations are given in Appendix B

```
> hv[1]:=z[1]*(-1+2*z[1]);
```

$$hv_1 := z_1\,(-1 + 2\,z_1)$$

```
> hv[2]:=z[2]*(-1+2*z[2]);
```

$$hv_2 := z_2\,(-1 + 2\,z_2)$$

```
> hv[3]:=(1-z[1]-z[2])*(-1+2*(1-z[1]-z[2]));
```

$$hv_3 := (1 - z_1 - z_2)\,(1 - 2\,z_1 - 2\,z_2)$$

```
> hv[4]:=4*z[1]*z[2];
```

$$hv_4 := 4\,z_1\,z_2$$

```
> hv[5]:=4*z[2]*(1-z[1]-z[2]);
```

$$hv_5 := 4\,z_2\,(1 - z_1 - z_2)$$

```
> hv[6]:=4*z[1]*(1-z[1]-z[2]);
```

$$hv_6 := 4\,z_1\,(1 - z_1 - z_2)$$

For the **pressure nodes**: the reference elemental basis from (reference) Element 1 to 3 are expressed below.

```
> hp[1]:=z[1];
```

$$hp_1 := z_1$$

```
> hp[2]:=z[2];
```

$$hp_2 := z_2$$

```
> hp[3]:=1-z[1]-z[2];
```

$$hp_3 := 1 - z_1 - z_2$$

Set up loops for alpha, beta and n:

```
> for alpha from 1 by 1 to 3 do:
> for beta from 1 by 1 to 6 do:
> for n from 1 by 1 to 2 do:
> eq1:=hp[alpha];
> eq2:=diff(hv[beta],z[n]);
> eq3:=eq1*eq2;eq4:=int(eq3,z[1]=0..1-z[2]);
> gd[alpha,beta,n]:=int(eq4,z[2]=0..1);
> unassign('z');z:=array(1..2);
> od;od;od;
```

Read in the libraries for "C" for conversion of the outputs to "C" language, and save the values for $G^{\widehat{D}}_{\alpha\beta n}$ in both "C" and "FORTRAN" languages.

```
> readlib(C):
> C(gd,filename="I:/Thesis/Courses/CFM/Problem4/P4a/gdfile.c");
> fortran(gd,filename="I:/Thesis/Courses/CFM/Problem4/P4a/gdfile.f");
```

I.2 Output Variables for $G^{\widehat{D}}_{\alpha\beta n}$ (FORTRAN version)

$gd(1,1,1) = 1.E0/6.E0$	$gd(1,1,2) = 0$	$gd(1,2,1) = 0$	$gd(1,2,2) = 0$
$gd(1,3,1) = 0$	$gd(1,3,2) = 0$	$gd(1,4,1) = 1.E0/6.E0$	$gd(1,4,2) = 1.E0/3.E0$
$gd(1,5,1) = -1.E0/6.E0$	$gd(1,5,2) = 0$	$gd(1,6,1) = -1.E0/6.E0$	$gd(1,6,2) = -1.E0/3.E0$
$gd(2,1,1) = 0$	$gd(2,1,2) = 0$	$gd(2,2,1) = 0$	$gd(2,2,2) = 1.E0/6.E0$
$gd(2,3,1) = 0$	$gd(2,3,2) = 0$	$gd(2,4,1) = 1.E0/3.E0$	$gd(2,4,2) = 1.E0/6.E0$
$gd(2,5,1) = -1.E0/3.E0$	$gd(2,5,2) = -1.E0/6.E0$	$gd(2,6,1) = 0$	$gd(2,6,2) = -1.E0/6.E0$
$gd(3,1,1) = 0$	$gd(3,1,2) = 0$	$gd(3,2,1) = 0$	$gd(3,2,2) = 0$
$gd(3,3,1) = -1.E0/6.E0$	$gd(3,3,2) = -1.E0/6.E0$	$gd(3,4,1) = 1.E0/6.E0$	$gd(3,4,2) = 1.E0/6.E0$
$gd(3,5,1) = -1.E0/6.E0$	$gd(3,5,2) = 1.E0/6.E0$	$gd(3,6,1) = 1.E0/6.E0$	$gd(3,6,2) = -1.E0/6.E0$

Appendix J

Derivation of $G^{\widehat{C}}_{\alpha\beta\delta\gamma}$ in Calculating $\left(\widehat{C}^k_{\alpha\delta}\right)^n$

J.1 MAPLE script

This MAPLE script calculates the $G^{\widehat{C}}_{\alpha\beta\delta\gamma}$ term in calculating $\left(\widehat{C}^k_{\alpha\delta}\right)^n$ term for convection in the Navier-Stokes problem.

*Note that the area multiplier (i.e., "2*area^k" where k is the element number) will be accounted for in the Stiffness Matrix*

Clear the variables from memory:

```
> restart;
```

Set the array for reference elemental axes z1 and z2 refer to ξ_1 and ξ_2, respectively

```
> z:=array(1..2);
```

$$z := \text{array}(1..2, [])$$

Set the array for the reference elemental basis. For linear triangular elements there are 3 nodes, and the array size is set from 1 to 3. However, for this second problem, quadratic triangular elements are used, with nodes between the corner elements. Therefore, for quadratic triangular elements, the array size for the reference elemental basis is 6, due to the 6 nodes for the element.

```
> h:=array(1..6);
```

$$h := \text{array}(1..6,\ [])$$

The G term in the Stiffness Matrix is a multidimensional array with the following dimensions:

```
> gcterm:=array(1..6,1..6,1..6,1..2);
```

$$gcterm := \text{array}(1..6,\ 1..6,\ 1..6,\ 1..2,\ [])$$

The reference elemental basis from Element 1 to 6 are expressed below. Verification of these correlations are given in Appendix B.

```
> h[1]:=z[1]*(-1+2*z[1]);
```

$$h_1 := z_1\,(-1 + 2\,z_1)$$

```
> h[2]:=z[2]*(-1+2*z[2]);
```

$$h_2 := z_2\,(-1 + 2\,z_2)$$

```
> h[3]:=(1-z[1]-z[2])*(-1+2*(1-z[1]-z[2]));
```

$$h_3 := (1 - z_1 - z_2)\,(1 - 2\,z_1 - 2\,z_2)$$

```
> h[4]:=4*z[1]*z[2];
```

$$h_4 := 4\,z_1\,z_2$$

```
> h[5]:=4*z[2]*(1-z[1]-z[2]);
```

$$h_5 := 4\,z_2\,(1 - z_1 - z_2)$$

```
> h[6]:=4*z[1]*(1-z[1]-z[2]);
```

$$h_6 := 4\,z_1\,(1 - z_1 - z_2)$$

Set up loops for alpha, beta, delta and gamma (used as gama to avoid name conflict):

```
> for alpha from 1 by 1 to 6 do:
> for beta from 1 by 1 to 6 do:
> for delta from 1 by 1 to 6 do:
> for gama from 1 by 1 to 2 do:
> eq1:=h[alpha];
> eq2:=h[beta];
```

```
> eq3:=diff(h[delta],z[gama]);

> eq4:=eq1*eq2*eq3; eq5:=int(eq4,z[1]=0..1-z[2]);

> gcterm[alpha,beta,delta,gama]:=int(eq5,z[2]=0..1);

> unassign('z');z:=array(1..2);

> od;od;od;od;
```

Read in the libraries for "C" for conversion of the outputs to "C" language, and save the values for GC in both "C" and "FORTRAN" languages.

Note that the order is G_{alpha, beta, delta, gamma) !

```
> readlib(C):

> C(gcterm,filename="I:/Thesis/Courses/CFM/Problem5/P5a/gcterm.c");
> fortran(gcterm,filename="I:/Thesis/Courses/CFM/Problem5/P5a/gcterm.f"
> );
```

J.2 Output Variables for $G^{\widehat{C}}_{\alpha\beta\delta\gamma}$

```
gcterm(1,1,1,1) = 13.E0/420.E0
gcterm(1,1,1,2) = 0
gcterm(1,1,2,1) = 0
gcterm(1,1,2,2) = -1.E0/140.E0
gcterm(1,1,3,1) = 1.E0/140.E0
gcterm(1,1,3,2) = 1.E0/140.E0
gcterm(1,1,4,1) = 1.E0/105.E0
gcterm(1,1,4,2) = 1.E0/21.E0
gcterm(1,1,5,1) = -1.E0/105.E0
gcterm(1,1,5,2) = 0
gcterm(1,1,6,1) = -4.E0/105.E0
gcterm(1,1,6,2) = -1.E0/21.E0
gcterm(1,2,1,1) = -1.E0/280.E0
gcterm(1,2,1,2) = 0
gcterm(1,2,2,1) = 0
gcterm(1,2,2,2) = -1.E0/280.E0
gcterm(1,2,3,1) = -11.E0/2520.E0
gcterm(1,2,3,2) = -11.E0/2520.E0
gcterm(1,2,4,1) = -2.E0/315.E0
gcterm(1,2,4,2) = -2.E0/315.E0
gcterm(1,2,5,1) = 2.E0/315.E0
gcterm(1,2,5,2) = 1.E0/126.E0
gcterm(1,2,6,1) = 1.E0/126.E0
gcterm(1,2,6,2) = 2.E0/315.E0
gcterm(1,3,1,1) = -1.E0/280.E0
gcterm(1,3,1,2) = 0
gcterm(1,3,2,1) = 0
gcterm(1,3,2,2) = 11.E0/2520.E0
gcterm(1,3,3,1) = 1.E0/280.E0
gcterm(1,3,3,2) = 1.E0/280.E0
gcterm(1,3,4,1) = 1.E0/630.E0
gcterm(1,3,4,2) = -2.E0/315.E0
gcterm(1,3,5,1) = -1.E0/630.E0
gcterm(1,3,5,2) = -1.E0/126.E0
gcterm(1,3,6,1) = 0
gcterm(1,3,6,2) = 2.E0/315.E0
gcterm(1,4,1,1) = 2.E0/105.E0
gcterm(1,4,1,2) = 0
gcterm(1,4,2,1) = 0
gcterm(1,4,2,2) = -4.E0/315.E0
gcterm(1,4,3,1) = 2.E0/315.E0
gcterm(1,4,3,2) = 2.E0/315.E0
gcterm(1,4,4,1) = -4.E0/315.E0
gcterm(1,4,4,2) = 2.E0/105.E0
gcterm(1,4,5,1) = 4.E0/315.E0
gcterm(1,4,5,2) = 2.E0/315.E0
gcterm(1,4,6,1) = -8.E0/315.E0
gcterm(1,4,6,2) = -2.E0/105.E0
gcterm(1,5,1,1) = 1.E0/210.E0
gcterm(1,5,1,2) = 0
gcterm(1,5,2,1) = 0
gcterm(1,5,2,2) = -1.E0/126.E0
gcterm(1,5,3,1) = 1.E0/126.E0
gcterm(1,5,3,2) = 1.E0/126.E0
gcterm(1,5,4,1) = -2.E0/105.E0
gcterm(1,5,4,2) = -2.E0/315.E0
gcterm(1,5,5,1) = 2.E0/105.E0
gcterm(1,5,5,2) = 0
gcterm(1,5,6,1) = -4.E0/315.E0
gcterm(1,5,6,2) = 2.E0/315.E0
gcterm(1,6,1,1) = 2.E0/105.E0
gcterm(1,6,1,2) = 0
gcterm(1,6,2,1) = 0
gcterm(1,6,2,2) = -2.E0/315.E0
gcterm(1,6,3,1) = 4.E0/315.E0
gcterm(1,6,3,2) = 4.E0/315.E0
gcterm(1,6,4,1) = -2.E0/315.E0
gcterm(1,6,4,2) = 2.E0/105.E0
gcterm(1,6,5,1) = 2.E0/315.E0
gcterm(1,6,5,2) = -2.E0/315.E0
gcterm(1,6,6,1) = -2.E0/63.E0
gcterm(1,6,6,2) = -2.E0/105.E0
gcterm(2,1,1,1) = -1.E0/280.E0
gcterm(2,1,1,2) = 0
gcterm(2,1,2,1) = 0
gcterm(2,1,2,2) = -1.E0/280.E0
gcterm(2,1,3,1) = -11.E0/2520.E0
gcterm(2,1,3,2) = -11.E0/2520.E0
gcterm(2,1,4,1) = -2.E0/315.E0
gcterm(2,1,4,2) = -2.E0/315.E0
gcterm(2,1,5,1) = 2.E0/315.E0
gcterm(2,1,5,2) = 1.E0/126.E0
gcterm(2,1,6,1) = 1.E0/126.E0
gcterm(2,1,6,2) = 2.E0/315.E0
gcterm(2,2,1,1) = -1.E0/140.E0
gcterm(2,2,1,2) = 0
gcterm(2,2,2,1) = 0
gcterm(2,2,2,2) = 13.E0/420.E0
gcterm(2,2,3,1) = 1.E0/140.E0
gcterm(2,2,3,2) = 1.E0/140.E0
gcterm(2,2,4,1) = 1.E0/21.E0
gcterm(2,2,4,2) = 1.E0/105.E0
gcterm(2,2,5,1) = -1.E0/21.E0
gcterm(2,2,5,2) = -4.E0/105.E0
gcterm(2,2,6,1) = 0
gcterm(2,2,6,2) = -1.E0/105.E0
gcterm(2,3,1,1) = 11.E0/2520.E0
gcterm(2,3,1,2) = 0
gcterm(2,3,2,1) = 0
gcterm(2,3,2,2) = -1.E0/280.E0
gcterm(2,3,3,1) = 1.E0/280.E0
gcterm(2,3,3,2) = 1.E0/280.E0
gcterm(2,3,4,1) = -2.E0/315.E0
gcterm(2,3,4,2) = 1.E0/630.E0
gcterm(2,3,5,1) = 2.E0/315.E0
gcterm(2,3,5,2) = 0
gcterm(2,3,6,1) = -1.E0/126.E0
gcterm(2,3,6,2) = -1.E0/630.E0
gcterm(2,4,1,1) = -4.E0/315.E0
gcterm(2,4,1,2) = 0
gcterm(2,4,2,1) = 0
gcterm(2,4,2,2) = 2.E0/105.E0
gcterm(2,4,3,1) = 2.E0/315.E0
gcterm(2,4,3,2) = 2.E0/315.E0
gcterm(2,4,4,1) = 2.E0/105.E0
gcterm(2,4,4,2) = -4.E0/315.E0
gcterm(2,4,5,1) = -2.E0/105.E0
gcterm(2,4,5,2) = -8.E0/315.E0
gcterm(2,4,6,1) = 2.E0/315.E0
gcterm(2,4,6,2) = 4.E0/315.E0
gcterm(2,5,1,1) = -2.E0/315.E0
gcterm(2,5,1,2) = 0
```

```
gcterm(2,5,2,1) = 0
gcterm(2,5,2,2) = 2.E0/105.E0
gcterm(2,5,3,1) = 4.E0/315.E0
gcterm(2,5,3,2) = 4.E0/315.E0
gcterm(2,5,4,1) = 2.E0/105.E0
gcterm(2,5,4,2) = -2.E0/315.E0
gcterm(2,5,5,1) = -2.E0/105.E0
gcterm(2,5,5,2) = -2.E0/63.E0
gcterm(2,5,6,1) = -2.E0/315.E0
gcterm(2,5,6,2) = 2.E0/315.E0
gcterm(2,6,1,1) = -1.E0/126.E0
gcterm(2,6,1,2) = 0
gcterm(2,6,2,1) = 0
gcterm(2,6,2,2) = 1.E0/210.E0
gcterm(2,6,3,1) = 1.E0/126.E0
gcterm(2,6,3,2) = 1.E0/126.E0
gcterm(2,6,4,1) = -2.E0/315.E0
gcterm(2,6,4,2) = -2.E0/105.E0
gcterm(2,6,5,1) = 2.E0/315.E0
gcterm(2,6,5,2) = -4.E0/315.E0
gcterm(2,6,6,1) = 0
gcterm(2,6,6,2) = 2.E0/105.E0
gcterm(3,1,1,1) = -1.E0/280.E0
gcterm(3,1,1,2) = 0
gcterm(3,1,2,1) = 0
gcterm(3,1,2,2) = 11.E0/2520.E0
gcterm(3,1,3,1) = 1.E0/280.E0
gcterm(3,1,3,2) = 1.E0/280.E0
gcterm(3,1,4,1) = 1.E0/630.E0
gcterm(3,1,4,2) = -2.E0/315.E0
gcterm(3,1,5,1) = -1.E0/630.E0
gcterm(3,1,5,2) = -1.E0/126.E0
gcterm(3,1,6,1) = 0
gcterm(3,1,6,2) = 2.E0/315.E0
gcterm(3,2,1,1) = 11.E0/2520.E0
gcterm(3,2,1,2) = 0
gcterm(3,2,2,1) = 0
gcterm(3,2,2,2) = -1.E0/280.E0
gcterm(3,2,3,1) = 1.E0/280.E0
gcterm(3,2,3,2) = 1.E0/280.E0
gcterm(3,2,4,1) = -2.E0/315.E0
gcterm(3,2,4,2) = 1.E0/630.E0
gcterm(3,2,5,1) = 2.E0/315.E0
gcterm(3,2,5,2) = 0
gcterm(3,2,6,1) = -1.E0/126.E0
gcterm(3,2,6,2) = -1.E0/630.E0
gcterm(3,3,1,1) = -1.E0/140.E0
gcterm(3,3,1,2) = 0
gcterm(3,3,2,1) = 0
gcterm(3,3,2,2) = -1.E0/140.E0
gcterm(3,3,3,1) = -13.E0/420.E0
gcterm(3,3,3,2) = -13.E0/420.E0
gcterm(3,3,4,1) = 1.E0/105.E0
gcterm(3,3,4,2) = 1.E0/105.E0
gcterm(3,3,5,1) = -1.E0/105.E0
gcterm(3,3,5,2) = 4.E0/105.E0
gcterm(3,3,6,1) = 4.E0/105.E0
gcterm(3,3,6,2) = -1.E0/105.E0
gcterm(3,4,1,1) = -1.E0/126.E0
gcterm(3,4,1,2) = 0
gcterm(3,4,2,1) = 0
gcterm(3,4,2,2) = -1.E0/126.E0
gcterm(3,4,3,1) = -1.E0/210.E0
gcterm(3,4,3,2) = -1.E0/210.E0
gcterm(3,4,4,1) = -2.E0/105.E0
gcterm(3,4,4,2) = -2.E0/105.E0
gcterm(3,4,5,1) = 2.E0/105.E0
gcterm(3,4,5,2) = 4.E0/315.E0
gcterm(3,4,6,1) = 4.E0/315.E0
gcterm(3,4,6,2) = 2.E0/105.E0
gcterm(3,5,1,1) = -2.E0/315.E0
gcterm(3,5,1,2) = 0
gcterm(3,5,2,1) = 0
gcterm(3,5,2,2) = -4.E0/315.E0
gcterm(3,5,3,1) = -2.E0/105.E0
gcterm(3,5,3,2) = -2.E0/105.E0
gcterm(3,5,4,1) = -4.E0/315.E0
gcterm(3,5,4,2) = -2.E0/315.E0
gcterm(3,5,5,1) = 4.E0/315.E0
gcterm(3,5,5,2) = 2.E0/63.E0
gcterm(3,5,6,1) = 8.E0/315.E0
gcterm(3,5,6,2) = 2.E0/315.E0
gcterm(3,6,1,1) = -4.E0/315.E0
gcterm(3,6,1,2) = 0
gcterm(3,6,2,1) = 0
gcterm(3,6,2,2) = -2.E0/315.E0
gcterm(3,6,3,1) = -2.E0/105.E0
gcterm(3,6,3,2) = -2.E0/105.E0
gcterm(3,6,4,1) = -2.E0/315.E0
gcterm(3,6,4,2) = -4.E0/315.E0
gcterm(3,6,5,1) = 2.E0/315.E0
gcterm(3,6,5,2) = 8.E0/315.E0
gcterm(3,6,6,1) = 2.E0/63.E0
gcterm(3,6,6,2) = 4.E0/315.E0
gcterm(4,1,1,1) = 2.E0/105.E0
gcterm(4,1,1,2) = 0
gcterm(4,1,2,1) = 0
gcterm(4,1,2,2) = -4.E0/315.E0
gcterm(4,1,3,1) = 2.E0/315.E0
gcterm(4,1,3,2) = 2.E0/315.E0
gcterm(4,1,4,1) = -4.E0/315.E0
gcterm(4,1,4,2) = 2.E0/105.E0
gcterm(4,1,5,1) = 4.E0/315.E0
gcterm(4,1,5,2) = 2.E0/315.E0
gcterm(4,1,6,1) = -8.E0/315.E0
gcterm(4,1,6,2) = -2.E0/105.E0
gcterm(4,2,1,1) = -4.E0/315.E0
gcterm(4,2,1,2) = 0
gcterm(4,2,2,1) = 0
gcterm(4,2,2,2) = 2.E0/105.E0
gcterm(4,2,3,1) = 2.E0/315.E0
gcterm(4,2,3,2) = 2.E0/315.E0
gcterm(4,2,4,1) = 2.E0/105.E0
gcterm(4,2,4,2) = -4.E0/315.E0
gcterm(4,2,5,1) = -2.E0/105.E0
gcterm(4,2,5,2) = -8.E0/315.E0
gcterm(4,2,6,1) = 2.E0/315.E0
gcterm(4,2,6,2) = 4.E0/315.E0
gcterm(4,3,1,1) = -1.E0/126.E0
gcterm(4,3,1,2) = 0
gcterm(4,3,2,1) = 0
gcterm(4,3,2,2) = -1.E0/126.E0
gcterm(4,3,3,1) = -1.E0/210.E0
gcterm(4,3,3,2) = -1.E0/210.E0
```

```
gcterm(4,3,4,1) = -2.E0/105.E0
gcterm(4,3,4,2) = -2.E0/105.E0
gcterm(4,3,5,1) = 2.E0/105.E0
gcterm(4,3,5,2) = 4.E0/315.E0
gcterm(4,3,6,1) = 4.E0/315.E0
gcterm(4,3,6,2) = 2.E0/105.E0
gcterm(4,4,1,1) = 4.E0/63.E0
gcterm(4,4,1,2) = 0
gcterm(4,4,2,1) = 0
gcterm(4,4,2,2) = 4.E0/63.E0
gcterm(4,4,3,1) = 4.E0/105.E0
gcterm(4,4,3,2) = 4.E0/105.E0
gcterm(4,4,4,1) = 16.E0/105.E0
gcterm(4,4,4,2) = 16.E0/105.E0
gcterm(4,4,5,1) = -16.E0/105.E0
gcterm(4,4,5,2) = -32.E0/315.E0
gcterm(4,4,6,1) = -32.E0/315.E0
gcterm(4,4,6,2) = -16.E0/105.E0
gcterm(4,5,1,1) = 2.E0/315.E0
gcterm(4,5,1,2) = 0
gcterm(4,5,2,1) = 0
gcterm(4,5,2,2) = 2.E0/63.E0
gcterm(4,5,3,1) = -2.E0/315.E0
gcterm(4,5,3,2) = -2.E0/315.E0
gcterm(4,5,4,1) = 8.E0/105.E0
gcterm(4,5,4,2) = 16.E0/315.E0
gcterm(4,5,5,1) = -8.E0/105.E0
gcterm(4,5,5,2) = -8.E0/315.E0
gcterm(4,5,6,1) = 0
gcterm(4,5,6,2) = -16.E0/315.E0
gcterm(4,6,1,1) = 2.E0/63.E0
gcterm(4,6,1,2) = 0
gcterm(4,6,2,1) = 0
gcterm(4,6,2,2) = 2.E0/315.E0
gcterm(4,6,3,1) = -2.E0/315.E0
gcterm(4,6,3,2) = -2.E0/315.E0
gcterm(4,6,4,1) = 16.E0/315.E0
gcterm(4,6,4,2) = 8.E0/105.E0
gcterm(4,6,5,1) = -16.E0/315.E0
gcterm(4,6,5,2) = 0
gcterm(4,6,6,1) = -8.E0/315.E0
gcterm(4,6,6,2) = -8.E0/105.E0
gcterm(5,1,1,1) = 1.E0/210.E0
gcterm(5,1,1,2) = 0
gcterm(5,1,2,1) = 0
gcterm(5,1,2,2) = -1.E0/126.E0
gcterm(5,1,3,1) = 1.E0/126.E0
gcterm(5,1,3,2) = 1.E0/126.E0
gcterm(5,1,4,1) = -2.E0/105.E0
gcterm(5,1,4,2) = -2.E0/315.E0
gcterm(5,1,5,1) = 2.E0/105.E0
gcterm(5,1,5,2) = 0
gcterm(5,1,6,1) = -4.E0/315.E0
gcterm(5,1,6,2) = 2.E0/315.E0
gcterm(5,2,1,1) = -2.E0/315.E0
gcterm(5,2,1,2) = 0
gcterm(5,2,2,1) = 0
gcterm(5,2,2,2) = 2.E0/105.E0
gcterm(5,2,3,1) = 4.E0/315.E0
gcterm(5,2,3,2) = 4.E0/315.E0
gcterm(5,2,4,1) = 2.E0/105.E0
gcterm(5,2,4,2) = -2.E0/315.E0
gcterm(5,2,5,1) = -2.E0/105.E0
gcterm(5,2,5,2) = -2.E0/63.E0
gcterm(5,2,6,1) = -2.E0/315.E0
gcterm(5,2,6,2) = 2.E0/315.E0
gcterm(5,3,1,1) = -2.E0/315.E0
gcterm(5,3,1,2) = 0
gcterm(5,3,2,1) = 0
gcterm(5,3,2,2) = -4.E0/315.E0
gcterm(5,3,3,1) = -2.E0/105.E0
gcterm(5,3,3,2) = -2.E0/105.E0
gcterm(5,3,4,1) = -4.E0/315.E0
gcterm(5,3,4,2) = -2.E0/315.E0
gcterm(5,3,5,1) = 4.E0/315.E0
gcterm(5,3,5,2) = 2.E0/63.E0
gcterm(5,3,6,1) = 8.E0/315.E0
gcterm(5,3,6,2) = 2.E0/315.E0
gcterm(5,4,1,1) = 2.E0/315.E0
gcterm(5,4,1,2) = 0
gcterm(5,4,2,1) = 0
gcterm(5,4,2,2) = 2.E0/63.E0
gcterm(5,4,3,1) = -2.E0/315.E0
gcterm(5,4,3,2) = -2.E0/315.E0
gcterm(5,4,4,1) = 8.E0/105.E0
gcterm(5,4,4,2) = 16.E0/315.E0
gcterm(5,4,5,1) = -8.E0/105.E0
gcterm(5,4,5,2) = -8.E0/315.E0
gcterm(5,4,6,1) = 0
gcterm(5,4,6,2) = -16.E0/315.E0
gcterm(5,5,1,1) = -4.E0/105.E0
gcterm(5,5,1,2) = 0
gcterm(5,5,2,1) = 0
gcterm(5,5,2,2) = 4.E0/63.E0
gcterm(5,5,3,1) = -4.E0/63.E0
gcterm(5,5,3,2) = -4.E0/63.E0
gcterm(5,5,4,1) = 16.E0/105.E0
gcterm(5,5,4,2) = 16.E0/315.E0
gcterm(5,5,5,1) = -16.E0/105.E0
gcterm(5,5,5,2) = 0
gcterm(5,5,6,1) = 32.E0/315.E0
gcterm(5,5,6,2) = -16.E0/315.E0
gcterm(5,6,1,1) = 2.E0/315.E0
gcterm(5,6,1,2) = 0
gcterm(5,6,2,1) = 0
gcterm(5,6,2,2) = 2.E0/315.E0
gcterm(5,6,3,1) = -2.E0/63.E0
gcterm(5,6,3,2) = -2.E0/63.E0
gcterm(5,6,4,1) = 16.E0/315.E0
gcterm(5,6,4,2) = 16.E0/315.E0
gcterm(5,6,5,1) = -16.E0/315.E0
gcterm(5,6,5,2) = 8.E0/315.E0
gcterm(5,6,6,1) = 8.E0/315.E0
gcterm(5,6,6,2) = -16.E0/315.E0
gcterm(6,1,1,1) = 2.E0/105.E0
gcterm(6,1,1,2) = 0
gcterm(6,1,2,1) = 0
gcterm(6,1,2,2) = -2.E0/315.E0
gcterm(6,1,3,1) = 4.E0/315.E0
gcterm(6,1,3,2) = 4.E0/315.E0
gcterm(6,1,4,1) = -2.E0/315.E0
gcterm(6,1,4,2) = 2.E0/105.E0
gcterm(6,1,5,1) = 2.E0/315.E0
gcterm(6,1,5,2) = -2.E0/315.E0
```

```
gcterm(6,1,6,1) = -2.E0/63.E0
gcterm(6,1,6,2) = -2.E0/105.E0
gcterm(6,2,1,1) = -1.E0/126.E0
gcterm(6,2,1,2) = 0
gcterm(6,2,2,1) = 0
gcterm(6,2,2,2) = 1.E0/210.E0
gcterm(6,2,3,1) = 1.E0/126.E0
gcterm(6,2,3,2) = 1.E0/126.E0
gcterm(6,2,4,1) = -2.E0/315.E0
gcterm(6,2,4,2) = -2.E0/105.E0
gcterm(6,2,5,1) = 2.E0/315.E0
gcterm(6,2,5,2) = -4.E0/315.E0
gcterm(6,2,6,1) = 0
gcterm(6,2,6,2) = 2.E0/105.E0
gcterm(6,3,1,1) = -4.E0/315.E0
gcterm(6,3,1,2) = 0
gcterm(6,3,2,1) = 0
gcterm(6,3,2,2) = -2.E0/315.E0
gcterm(6,3,3,1) = -2.E0/105.E0
gcterm(6,3,3,2) = -2.E0/105.E0
gcterm(6,3,4,1) = -2.E0/315.E0
gcterm(6,3,4,2) = -4.E0/315.E0
gcterm(6,3,5,1) = 2.E0/315.E0
gcterm(6,3,5,2) = 8.E0/315.E0
gcterm(6,3,6,1) = 2.E0/63.E0
gcterm(6,3,6,2) = 4.E0/315.E0
gcterm(6,4,1,1) = 2.E0/63.E0
gcterm(6,4,1,2) = 0
gcterm(6,4,2,1) = 0
gcterm(6,4,2,2) = 2.E0/315.E0
gcterm(6,4,3,1) = -2.E0/315.E0
gcterm(6,4,3,2) = -2.E0/315.E0
gcterm(6,4,4,1) = 16.E0/315.E0
gcterm(6,4,4,2) = 8.E0/105.E0
gcterm(6,4,5,1) = -16.E0/315.E0
gcterm(6,4,5,2) = 0
gcterm(6,4,6,1) = -8.E0/315.E0
gcterm(6,4,6,2) = -8.E0/105.E0
gcterm(6,5,1,1) = 2.E0/315.E0
gcterm(6,5,1,2) = 0
gcterm(6,5,2,1) = 0
gcterm(6,5,2,2) = 2.E0/315.E0
gcterm(6,5,3,1) = -2.E0/63.E0
gcterm(6,5,3,2) = -2.E0/63.E0
gcterm(6,5,4,1) = 16.E0/315.E0
gcterm(6,5,4,2) = 16.E0/315.E0
gcterm(6,5,5,1) = -16.E0/315.E0
gcterm(6,5,5,2) = 8.E0/315.E0
gcterm(6,5,6,1) = 8.E0/315.E0
gcterm(6,5,6,2) = -16.E0/315.E0
gcterm(6,6,1,1) = 4.E0/63.E0
gcterm(6,6,1,2) = 0
gcterm(6,6,2,1) = 0
gcterm(6,6,2,2) = -4.E0/105.E0
gcterm(6,6,3,1) = -4.E0/63.E0
gcterm(6,6,3,2) = -4.E0/63.E0
gcterm(6,6,4,1) = 16.E0/315.E0
gcterm(6,6,4,2) = 16.E0/105.E0
gcterm(6,6,5,1) = -16.E0/315.E0
gcterm(6,6,5,2) = 32.E0/315.E0
gcterm(6,6,6,1) = 0
gcterm(6,6,6,2) = -16.E0/105.E0
```

Appendix K

Memory gain from sparse matrix usage

K.1 Global Stiffness and Mass Matrices

The `first step` in assembling the system matrix is *assembling the global stiffness matrix* ($\widetilde{A}$) from the elemental stiffness matrices ($\hat{A}^k_{\alpha\beta}$), and the *gobal mass matrix* ($\hat{M}^k_{\alpha\beta}$) from the elemental mass matrices ($\hat{M}^k_{\alpha\beta}$). The size of $\widetilde{A}$ and $\widetilde{M}$ are ($\mathcal{M}^{\mathcal{V}} \times \mathcal{M}^{\mathcal{V}}$). The algorithm to assemble $\widetilde{A}$ and $\widetilde{M}$ is given below:

$$
\begin{aligned}
&For\ 1 \leq k \leq Number\ of\ elements\ (K)\\
&\quad \{For\ 1 \leq \alpha \leq\ (Number\ of\ local\ velocity\ nodes (= 6))\\
&\qquad \{i = (g^V)^k_\alpha\\
&\qquad\quad \{For\ 1 \leq \beta \leq\ (Number\ of\ local\ velocity\ nodes (= 6))\\
&\qquad\qquad \{j = (g^V)^k_\beta\\
&\qquad\qquad (\widetilde{A})_{ij} = (\widetilde{A})_{ij} + (\widehat{A})^k_{\alpha\beta}\\
&\qquad\qquad (\widetilde{M})_{ij} = (\widetilde{M})_{ij} + (\widehat{M})^k_{\alpha\beta}\\
&\quad \}\qquad\qquad \}\qquad \}
\end{aligned}
\tag{K.1}
$$

Two identical sums of global stiffness and mass matrices (see Equation 3 below) will form the upper section of the system matrix, with complementing zero matrices. In a way, these two

sums of matrices represent the velocity nodes for the x and y coordinate directions.

K.2 Global Gradient Matrices

The `second step` in assembling the system matrix is *assembling the global gradient matrices* ($\widetilde{D}_1$ and $\widetilde{D}_2$) from the elemental gradient matrices ($(\widehat{D}_i)^k_{\alpha\beta}$). The size of both $\widetilde{D}_1$ and $\widetilde{D}_2$ matrices are $(\mathcal{M}^{\mathcal{P}} \times \mathcal{M}^{\mathcal{V}})$. These global matrices will form the pressure related lower and right side sections of the system matrix (with supplementing zero matrices at the lower right part of the system matrix).

The assembly of $\widetilde{D}_i$ (i=1,2) is performed using the following algorithm (similar to the assembly of the global stiffness matrix). (The algorithm will be given for $\widetilde{D}_1$, while for $\widetilde{D}_2$ the same logic will apply):

$$
\begin{array}{l}
\text{zero } \widetilde{D}_1 \\
\\
For\ 1 \leq k \leq Number\ of\ elements\ (K) \\
\quad \{For\ 1 \leq \alpha \leq\ (Number\ of\ local\ pressure\ nodes (= 3)) \\
\quad\quad \{i = (g^P)^k_\alpha \\
\quad\quad\quad \{For\ 1 \leq \beta \leq\ (Number\ of\ local\ velocity\ nodes (= 6)) \\
\quad\quad\quad\quad \{j = (g^V)^k_\beta \\
\quad\quad\quad\quad (\widetilde{D}_1)_{mn} = (\widetilde{D}_1)_{mn} + (\widehat{D}_1)^k_{\alpha\beta} \\
\quad \} \qquad\quad \} \qquad \}
\end{array}
\tag{K.2}
$$

K.3 General Form of the System Matrix

Once the global stiffness matrix ($\widetilde{A}$) the global mass matrix ($\widetilde{M}$), and the global gradient matrices ($\widetilde{D}_1$ and $\widetilde{D}_2$) are obtained, the system matrix for the left hand side of the Navier-

Stokes equation can be assembled as:

$$\widetilde{S} = \left[\begin{array}{|c|c|c|} \hline \frac{1}{Re}\widetilde{A} + \frac{1}{\Delta t}\widetilde{M} & 0 & -\widetilde{D}_1^T \\ \hline 0 & \frac{1}{Re}\widetilde{A} + \frac{1}{\Delta t}\widetilde{M} & -\widetilde{D}_2^T \\ \hline -\widetilde{D}_1 & -\widetilde{D}_2 & 0 \\ \hline \end{array}\right] \tag{K.3}$$

The size of the system matrix is $[(\mathcal{M}^{\mathcal{V}} + \mathcal{M}^{\mathcal{V}} + \mathcal{M}^{\mathcal{P}}) \times (\mathcal{M}^{\mathcal{V}} + \mathcal{M}^{\mathcal{V}} + \mathcal{M}^{\mathcal{P}})]$, where $\mathcal{M}^{\mathcal{V}}$ and $\mathcal{M}^{\mathcal{P}}$) are the number of velocity and pressure nodes, respectively.

K.4 Sparse System Matrix

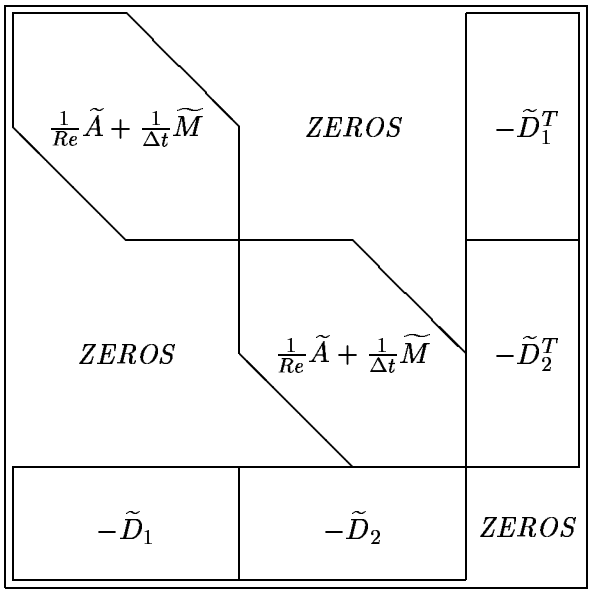
As mentioned in the previous paragraph, the full system matrix requires a storage of $(2\mathcal{M}^{\mathcal{V}} + \mathcal{M}^{\mathcal{P}})^2$. However, significant savings in storage space can be achieved by using sparse matrices. As shown in Figure K.1, large portions of the system matrix have zero element values. From those sections a memory saving of $2(\mathcal{M}^{\mathcal{V}})^2 + (\mathcal{M}^{\mathcal{P}})^2$ can be achieved. In addition to that, the stiffness and mass matrices are banded, which results in additional saving in storage memory.

This memory management can be implemented by gradual dynamic array allocation and deallocation. The process is explained here from the stage, where the elemental constructions are already created and available. These are the elemental stiffness (A_ESM), mass (MM_EMM) and gradient ($DD1\_EGM$ and $DD2\_EGM$) matrices.

The code follows the steps described below to minimize memory usage, consequently to increase the mesh size to be calculated:

1. `Allocate` memory to the global stiffness matrix (AG).

Figure K.1: Realized storage gain by using a sparse system matrix



2. Call up subroutine $GLOBAL\_A$ to calculate the global stiffness matrix(AG). If the Navier-Stokes option is selected, multiply the global stiffness matrix within the subroutine by (1/Re)

$$AG = \frac{1}{Re} \cdot AG$$

3. **Deallocate** the elemental stiffness matrix (A_ESM)

4. `Allocate` memory to the global mass matrix (MG).

5. If the Navier-Stokes option is selected, calculate the global mass matrix (MG) through the subroutine $GLOBAL\_MM$. Note that the Mass matrix is multiplied by ($1/\Delta t$)

$$MG = \frac{1}{\Delta t} \cdot MG$$

6. **Deallocate** the elemental stiffness matrix (MM_EMM)

7. If the Navier-Stokes options is selected, sum up the stiffness and mass matrices

$$AG = AG + MG$$

8. Calculate the number of non-zero elements in stiffness matrix AG and return the value – it is used to allocate memory for the sparse stiffness matrix.

9. `Allocate` memory to the sparse stiffness matrix.

10. *Sparse* the stiffness matrix into three sparse vectors. The sparse coordinate format for the matrix A requires one real and two integer vectors. The real array a contains all the nonzeros in A. Let the number of nonzeros be nz. The two integer arrays $irow$ and $jcol$, each of length nz, contain the row and column numbers for these entries in A. That is

$$A_{irow(i),jcol(i)} = a(i), \qquad i = 1, \ldots, nz$$

with all other entries in A zero.

11. **Deallocate** AG

12. Calculate the number of non-zero elements in the mass matrix MG and return the value – it is used to allocate memory for the sparse mass matrix

13. `Allocate` memory to the sparse mass matrix.

14. *Sparse* the mass matrix into three sparse vectors (see Point 8).

15. **Deallocate** MG

---

16. Call the subroutine $GLOBAL\_DD1$ to assemble the global gradient matrix

17. **Deallocate** the elemental gradient matrix $DD1\_EGM$

18. `Allocate` vector size to the Global Gradient matrix $DD1\_GLOBAL$, using one real and two integer vectors.

19. *Sparse* the global gradient matrix $DD1\_GLOBAL$ into one real and two integer vectors

20. **Deallocate** $DD1\_GLOBAL$

    .....

21. `Allocate` vector size to the transposed Global Gradient matrix $DD1\_GLOB\_TR$, using one real and two integer vectors.

22. *Sparse* the global transposed gradient matrix $DD1\_GLOB\_TR$ into one real and two integer vectors

23. **Deallocate** $DD1\_GLOB\_TR$

---

24. Call the subroutine $GLOBAL\_DD2$ to assemble the global gradient matrix

25. **Deallocate** the elemental gradient matrix $DD2\_EGM$

26. `Allocate` vector size to the Global Gradient matrix $DD2\_GLOBAL$, using one real and two integer vectors.

27. *Sparse* the global gradient matrix $DD2\_GLOBAL$ into one real and two integer vectors

28. **Deallocate** $DD2\_GLOBAL$

    .....

29. `Allocate` vector size to the transposed Global Gradient matrix $DD2\_GLOB\_TR$, using one real and two integer vectors.

30. *Sparse* the global transposed gradient matrix $DD2\_GLOB\_TR$ into one real and two integer vectors

31. **Deallocate** $DD2\_GLOB\_TR$

Once the left hand side is sparsed; the right hand side forcing term is assembled; and the appropriate rows and columns are removed due to initial and boundary conditions (including the last row and colums of the right hand side for the reference pressure), the right hand side is LU decomposed using the IMSL library routine: $DLFTXG$, and the system of equations is solved using the IMSL library routine: $DLFSXG$.

Appendix L

Algorithm for a conjugate gradient solver

This method ensures that the search directions are A-orthogonal. This eliminates the hem-stitching problem of the steepest-descent method.

Definition

The superscript in brackets represents the iteration number. $u^{(m)}$ is the solution vector, $r^{(m)}$ is the residual vector, $d^{(m)}$ is the new search direction vector and $a^{(m)}$ is the step size (scalar) at the m^{th} iteration step.

$$u^{(0)} = 0 \tag{L.1}$$

$$r^{(0)} = f - (Au)^{(0)} = f - 0 = f \tag{L.2}$$

At the first iteration step:

$$b^{(1)} = 0 \tag{L.3}$$

$$d^{(1)} = r^{(0)} \tag{L.4}$$

And for all $m = 1, ..., M$, where M is the number of iterations:

$$b^{(m)} = \frac{r^{(m-1)^T} r^{(m-1)}}{r^{(m-2)^T} r^{(m-2)}} \tag{L.5}$$

$$d^{(m)} = r^{(m-1)} + b^{(m)} d^{(m-1)} \tag{L.6}$$

$$a^{(m)} = \frac{r^{(m-1)^T} r^{(m-1)}}{d^{(m)^T} A\ d^{(m)}} \tag{L.7}$$

$$u^{(m)} = u^{(m-1)} + a^{(m)} d^{(m)} \tag{L.8}$$

$$r^{(m)} = r^{(m-1)} + a^{(m)} A\ d^{(m)} \tag{L.9}$$

Note that Equation 9 was derived from:

$$r^{(m)} = f - (Au)^{(m)} - (Au)^{(m-1)} + (Au)^{(m-1)} = r^{(m-1)} + a^{(m)} A\ d^{(m)} \tag{L.10}$$

Properties

The conjugate direction is:

$$d^{(i)^T} A\ d^{(j)} = 0 \qquad \forall\ i \neq j \tag{L.11}$$

$$r^{(i)^T} r^{(j)} = 0 \qquad \forall\ i \neq j \tag{L.12}$$

Finite termination:

$$u^{(M)} = u \tag{L.13}$$

where $u^{(M)}$ is the M^{th} iteration solution and u is the exact solution.

For M number of distinct eigenvalues of A, and for K number of elements:

$$M = \sqrt{\frac{K}{2}}\ ln(tolerance) \tag{L.14}$$

for $1D \sim$ O(1/h) or $1D \sim$ O(N)

Stopping criterion (FEM Matrix Free Implementation)

$$\begin{aligned} |u - u_h^{(M)}|_{H^1} &= |u - u_h + u_h - u_h^{(M)}|_{H^1} \\ &\leq \underbrace{|u - u_h|_{H^1}}_{\varepsilon_{discretization}} + \underbrace{|u_h - u_h^{(M)}|_{H^1}}_{\varepsilon_{iteration}} \end{aligned} \tag{L.15}$$

We want an absolute criterion that does not require knowledge of $K(A)$.

For the m^{th} iteration step:

$$\tilde{r}^{(m)} = \tilde{f} - \tilde{A}\tilde{u}^{(m)} \tag{L.16}$$

Stopping:

$$\left\{ \sum_{i=1}^{\#ofnodes} \left[\left(\tilde{r}_i^{(m)} \right)^2 \; B_i^{-1} \right] \right\}^{1/2} < \frac{\pi}{L} \quad \varepsilon_{iteration} \tag{L.17}$$

where L is the characteristic length of the domain (its maximum size), and B_i is a vector (with elements for each node), called the lumped mass matrix.

The lumped mass matrix (B_i) can be pre-calculated before the iterations start. It is useful, since B_i does not change through the iterations, and the iteration loop needs to calculate the stopping criteria when the program steps into the loop. (B_i) can be calculated from:

$$\begin{aligned}
&For\ 1 \leq k \leq K \\
&\quad \{For\ 1 \leq \alpha \leq\ (Number\ of\ local\ nodes) \\
&\qquad \{i = g_\alpha^k \\
&\qquad\quad B_i = B_i + \frac{area^k\ \widehat{M}_{\alpha\alpha}^k}{\sum_{\beta=1}^{(\#\ of\ local\ nodes)} \widehat{M}_{\beta\beta}^k} \\
&\quad \} \qquad \}
\end{aligned} \tag{L.18}$$

where K is the total number of elements, and i is the global node number, corresponding to a specific element (k) and local node (α). g_α^k is the local to global mapping, representing the k^{th} element and the α^{th} local node, and the equation assigns the corresponding global node number to i. For quadratic triangular elements, the number of local nodes is 6, while for linear triangular elements it is 3.

Initialization (FEM Matrix Free Implementation)

First initialize the forcing term vector to 0, for each of it vector elements (having as many elements as nodes in the domain.)

$$\tilde{f}_i = 0 \tag{L.19}$$

Then the forcing term can be calculated from:

$$\begin{aligned}
&For\ 1 \leq k \leq K \\
&\quad \{For\ 1 \leq \alpha \leq\ (Number\ of\ local\ nodes) \\
&\qquad \{i = g_\alpha^k \\
&\qquad\quad \{For\ 1 \leq \beta \leq\ (Number\ of\ local\ nodes) \\
&\qquad\qquad \{ \\
&\qquad\qquad \tilde{f}_i = \tilde{f}_i + \widehat{M}_{\alpha\beta}^k \cdot f(\hat{x}_\beta^k) \\
&\quad \} \qquad\quad \} \qquad \}
\end{aligned} \tag{L.20}$$

where K represents the number of elements, and $f(\hat{x}_\beta^k)$ is the value calculated from the "forcing correlation" using the x and y coordinates at the k^{th} element and β^{th} local node. (Note that the forcing correlation does not have to depend on the x and y global coordinates of the node.) For example, if the forcing correlation is $f = sin(x\pi)sin(y\pi)$, then $f(\hat{x}_3^7)$ means this: get the global x and y coordinates of Element 7, Local Node 3, substitute it into the correlation for f and calculate its value.

Once the forcing term is calculated, initialize the solution and residual vectors for the 0^{th} iteration. (These vectors have elements for each node.)

$$\tilde{u}^{(0)} = 0 \tag{L.21}$$

$$\tilde{r}^{(0)} = mask^T \cdot \tilde{f} \tag{L.22}$$

where the *mask* for the boundary conditions (i.e., nodal boundaries) is introduced as:

$$mask_i = \begin{cases} 0 & boundary\ nodes \\ 1 & interior\ nodes \end{cases} \tag{L.23}$$

Start Iterations (FEM Matrix Free Implementation)

Perform iterations for $m = 1, .., M$. Note that in the first iteration, the values for $b^{(1)}$, and $d^{(1)}$ are defined by Equations 3 and 4, respectively.

$$\gamma = \tilde{r}^{(m-1)^T} \cdot \tilde{r}^{(m-1)} \tag{L.24}$$

$$b^{(m)} = \frac{\gamma}{\gamma\,'} \tag{L.25}$$

$$\gamma\,' = \gamma \tag{L.26}$$

$$\tilde{d}^{(m)} = mask \cdot (\tilde{r}^{(m-1)} + b^{(m)} \tilde{d}^{(m-1)}) \tag{L.27}$$

Initialize the $\tilde{v}'$ vector to 0, for each of it vector elements (having as many elements as nodes in the domain.)

$$\tilde{v}_i\,' = 0 \tag{L.28}$$

Then the forcing term can be calculated from:

$$\begin{aligned} &For\ 1 \le k \le K \\ &\quad \{For\ 1 \le \alpha \le\ (Number\ of\ local\ nodes) \\ &\qquad \{i = g_\alpha^k \\ &\qquad\quad \{For\ 1 \le \beta \le\ (Number\ of\ local\ nodes) \\ &\qquad\qquad \{j = g_\beta^k \\ &\qquad\qquad \tilde{v}_i\,' = \tilde{v}_i\,' + \hat{A}_{\alpha\beta}^k \cdot d_j^{(m)} \\ &\quad \} \qquad\quad \} \qquad \} \end{aligned} \tag{L.29}$$

where $A_{\alpha\beta}^k$ is the elemental stiffness matrix for the k^{th} element.

From this calculate further:

$$\tilde{v} = mask \cdot \tilde{v}\,' \tag{L.30}$$

$$a^{(m)} = \frac{\gamma}{d^{(m)^T}\tilde{v}} \tag{L.31}$$

$$\tilde{u}^{(m)} = \tilde{u}^{(m-1)} + a^{(m)}\tilde{d}^{(m)} \tag{L.32}$$

$$\tilde{r}^{(m)} = mask \cdot (\tilde{r}^{(m-1)} - a^{(m)}\tilde{v}) \tag{L.33}$$

At this point re-evaluate the stopping criteria, and if it is not satisfied, perform a new iteration step.

Appendix M

Derivation of the finite-difference operator

Let us assume that the domain has 5 nodal points. The third node is identified as i. The nodes before and after are identified by the appropriate numbering, as shown in Figure M.1.

M.1 First derivative ($\partial w/\partial x$)

Between nodes $i-2$ and $i-1$:

$$\left.\frac{\partial w}{\partial x}\right|_{i-1.5} = \frac{w_{i-1} - w_{i-2}}{\delta x} \tag{M.1}$$

Between nodes $i-1$ and i:

$$\left.\frac{\partial w}{\partial x}\right|_{i-0.5} = \frac{w_{i} - w_{i-1}}{\delta x} \tag{M.2}$$

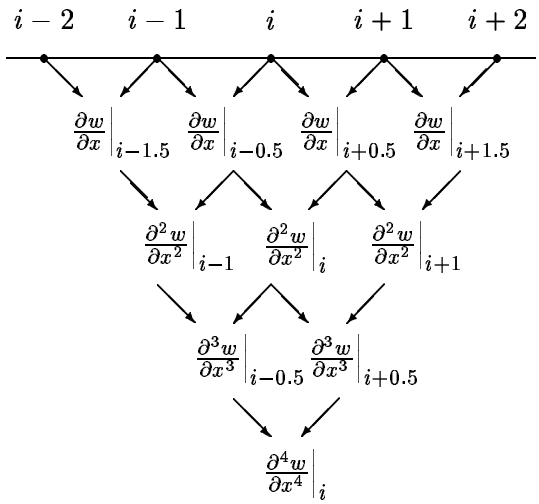
Between nodes i and $i+1$:

$$\left.\frac{\partial w}{\partial x}\right|_{i+0.5} = \frac{w_{i+1} - w_{i}}{\delta x} \tag{M.3}$$

Between nodes $i+1$ and $i+2$:

$$\left.\frac{\partial w}{\partial x}\right|_{i+1.5} = \frac{w_{i+2} - w_{i+1}}{\delta x} \tag{M.4}$$

Figure M.1: Visualization of the finite-difference operator derivativatives



M.2 Second derivative ($\partial^2 w/\partial x^2$)

At node $i-1$:

$$\left.\frac{\partial^2 w}{\partial x^2}\right|_{i-1} = \frac{\left.\frac{\partial w}{\partial x}\right|_{i-0.5} - \left.\frac{\partial w}{\partial x}\right|_{i-1.5}}{\partial x} = \frac{w_i - 2w_{i-1} + w_{i-2}}{\delta x^2} \tag{M.5}$$

At node i:

$$\left.\frac{\partial^2 w}{\partial x^2}\right|_{i} = \frac{\left.\frac{\partial w}{\partial x}\right|_{i+0.5} - \left.\frac{\partial w}{\partial x}\right|_{i-0.5}}{\partial x} = \frac{w_{i+1} - 2w_i + w_{i-1}}{\delta x^2} \tag{M.6}$$

At node $i+1$:

$$\left.\frac{\partial^2 w}{\partial x^2}\right|_{i+1} = \frac{\left.\frac{\partial w}{\partial x}\right|_{i+1.5} - \left.\frac{\partial w}{\partial x}\right|_{i+0.5}}{\partial x} = \frac{w_{i+2} - 2w_{i+1} + w_i}{\delta x^2} \tag{M.7}$$

M.3 Third derivative ($\partial^3 w/\partial x^3$)

Between nodes $i-1$ and i:

$$\left.\frac{\partial^3 w}{\partial x^3}\right|_{i-0.5} = \frac{\left.\frac{\partial^2 w}{\partial x^2}\right|_i - \left.\frac{\partial^2 w}{\partial x^2}\right|_{i-1}}{\partial x} = \frac{w_{i+1} - 3w_i + 3w_{i-1} - w_{i-2}}{\delta x^3} \tag{M.8}$$

Between nodes i and $i+1$:

$$\left.\frac{\partial^3 w}{\partial x^3}\right|_{i+0.5} = \frac{\left.\frac{\partial^2 w}{\partial x^2}\right|_{i+1} - \left.\frac{\partial^2 w}{\partial x^2}\right|_i}{\partial x} = \frac{w_{i+2} - 3w_{i+1} + 3w_i - w_{i-1}}{\delta x^3} \tag{M.9}$$

M.4 Fourth derivative ($\partial^4 w/\partial x^4$)

At node i:

$$\left.\frac{\partial^4 w}{\partial x^4}\right|_i = \frac{\left.\frac{\partial^3 w}{\partial x^3}\right|_{i+0.5} - \left.\frac{\partial^3 w}{\partial x^3}\right|_{i-0.5}}{\partial x} = \frac{w_{i+2} - 4w_{i+1} + 6w_i - 4w_{i-1} + w_{i-2}}{\delta x^4} \tag{M.10}$$

Appendix N

Additional geometries for duct flow

N.1 Poiseuille type duct flow

Steady flow in a cylindrical tube is described by the Poiseuille equation, which states that the pressure drop along the tube is directly proportional to the length of the tube, the rate of the flow and the viscosity of the fluid, and inversely proportional to the fourth power of the internal radius. If dye is injected into the tube the liquid in the axis of the tube moves much faster than that near the wall and the front of the dye assumes a parabolic shape. This occurs because particles of fluid are flowing in a series of laminae parallel to the sides of the tube and the fluid in contact with the wall is stationary and each successive lamina is slipping against the viscous friction of the lamina outside. When flow occurs in such parallel laminea it is called laminar.

In the present thesis the Navier–Stokes code was used to solve the flow equations inside a two-dimensional duct. The fluid entered the duct at a uniform velocity and due to the no slip boundary condition it developed a parabolic velocity profile by the time it left the duct at the opposite end. The duct was discretized to 2560 elements ($N_x = 80$, $N_y = 16$, with two triangular elements in each rectangular element block). The test matrix is given in Table N.1. A typical example of the results is shown in Figure N.1. The results are found to be in good agreement with the theoretical results.

Figure N.1: Sample result for 2D Poiseuille flow

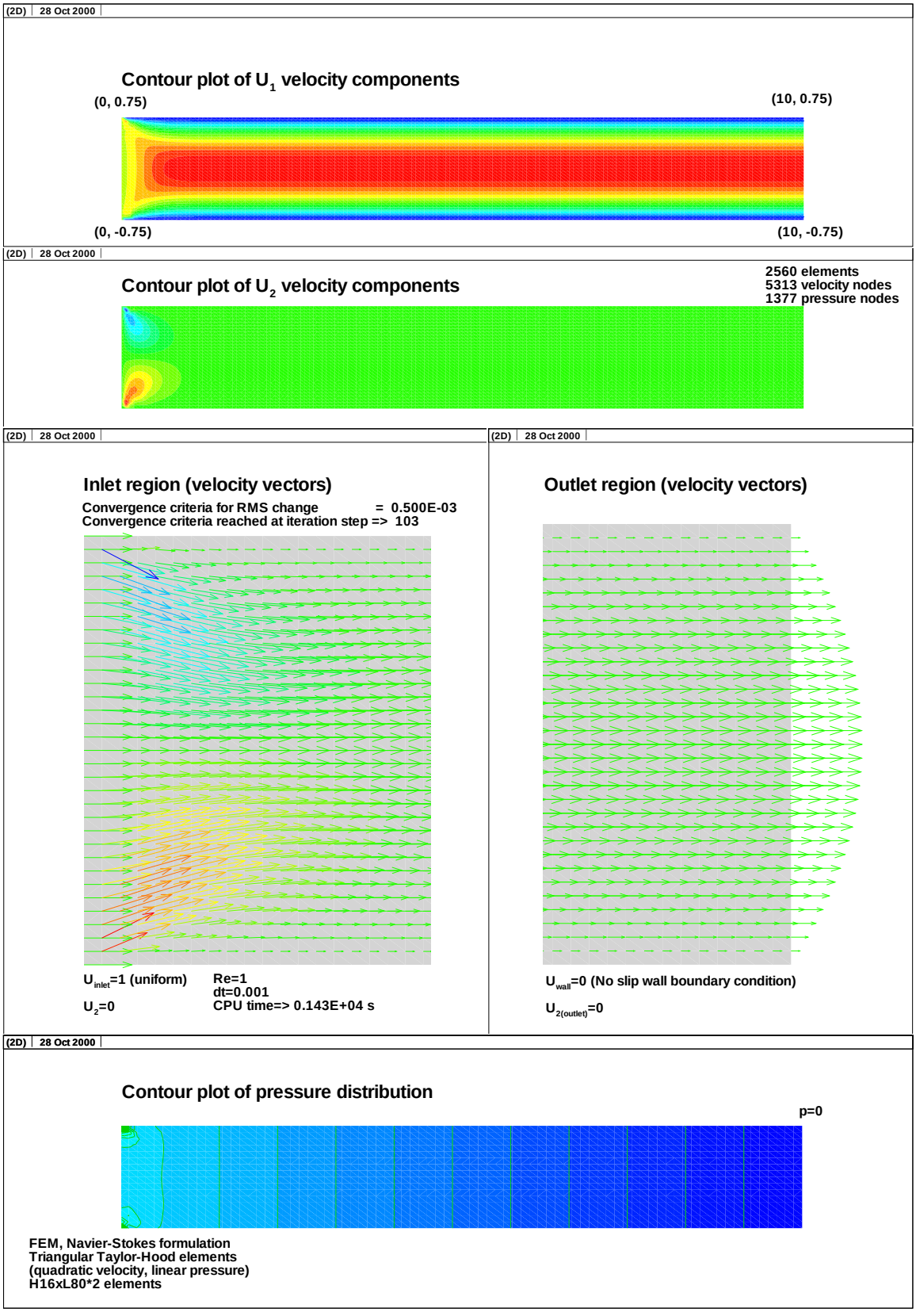


Table N.1: Test matrix for the duct flow cases

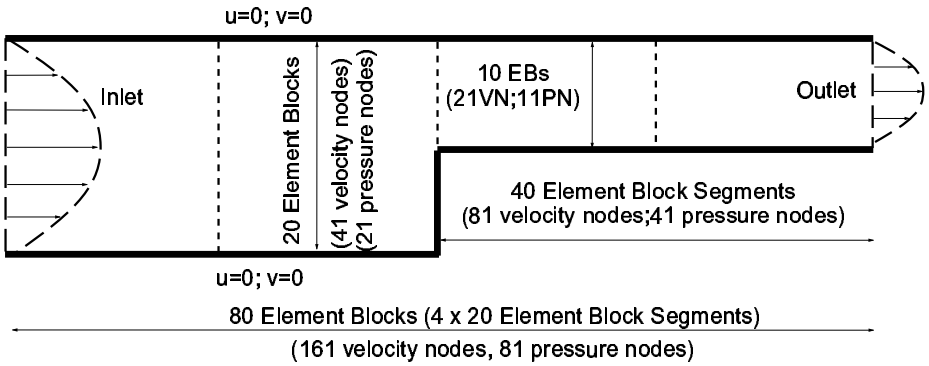
$Case\#$	$Flow\ rate$	$Velocity$	Re	dt_{crit}	dt_{used}	RMS_{crit}	$Iter\#$	CPU
$NS1$	0.00005	0.1592	189	0.00071	0.00035	0.00005	40	1267
$NS2$	0.0001	0.3183	378	0.00035	0.00018	0.00005	26	825
$NS2a$	0.0001	0.3183	378	0.00035	0.00018	0.00001	31	767
$NS3$	0.0002	0.6366	756	0.0002	0.00009	0.00001	92	429
$NS4$	0.0004	1.2732	1510	0.0001	0.00004	0.00001	297	537
$NS5$	0.0006	1.9099	2270	0.0001	0.00003	0.00001	595	929
$NS6$	0.0008	2.5465	3020	0.0000	0.00002	0.00001	949	1194
$NS7$	0.001	3.1831	3780	0.0000	0.00002	0.00001	1348	2188

N.2 Forward facing step

Perhaps the best test problems that retain the features of geometrically simple flow domains are the forward-facing step and the full step problems. The forward facing step is shown in Figure N.2. The distribution of boundary conditions is similar to that for the backward-facing step problem. These problems are realistic in a sense that they do not scale with the Reynolds-number and that meaningful experimental data can be used for comparison purposes. The step is set in the middle of the domain along the x-axis. This allows for the development of the upstream recirculation zone, the midstream detachment at the step face and the downstream reattachment as well. The domain length is set to $L = 0.06\ m$ ($0.08m$ and $0.04m$ for ST1a and NS7a cases, respectively). The duct height was set to $h = 0.01m$. Note that, at the inlet a uniform velocity profile was used, unlike the parabolic distributions shown in Figure N.2.

The axial length and duct height were divided into 80 and 20 element blocks, respectively. A 40 element long and 10 element high block was removed from the mesh to achieve the forward facing step geometry. This discretization resulted in a 2400 element domain (2600 elements for Case NS7a). The test conditions are given in Table N.2. A typical example of the results is shown in Figure N.3, for Re=189 and Re=3780. Both the low and the high Reynolds-number

Figure N.2: Forward facing step



cases exhibit a recirculation zone at the bottom corner of the forward facing step. At high Reynolds-number there is a small detachment and reattachment at the upper corner of the step. Due to the coarse mesh this detachment is not well pronounced. At low Reynolds-numbers the pressure distribution show cross sectional uniformity. This changes at high Reynolds-numbers, where the increased flow rate and flow distribution has upstream and downstream effects on the pressure field around the step. It should be noted that the critical time step should be based on the outlet conditions, since the velocity is the highest at that locations. Therefore, to achieve stability, a more conservative value was used in the calculations. The flow enters at the left side of the domain with uniform velocity. Due to the boundary conditions, at the exit the distribution is parabolic.

N.3 Vertical wall (fence) in a duct

The vertical wall is an extreme case of the full step configuration. As shown in Figure N.4, the step length is assumed virtually zero. This configuration maximizes the lengths of upstream and downstream re-circulation zones. The $L = 0.04m$ domain is divided into 64 element blocks, while the height ($h = 0.01m$) is divided into 20 element blocks. The 10 element high vertical fence is set up in the middle of the domain, 32 element blocks from the exit. For the cases where the Reynolds-number exceeded 3000 the fence was moved further upstream to allow for

Figure N.3: Sample result for forward facing step

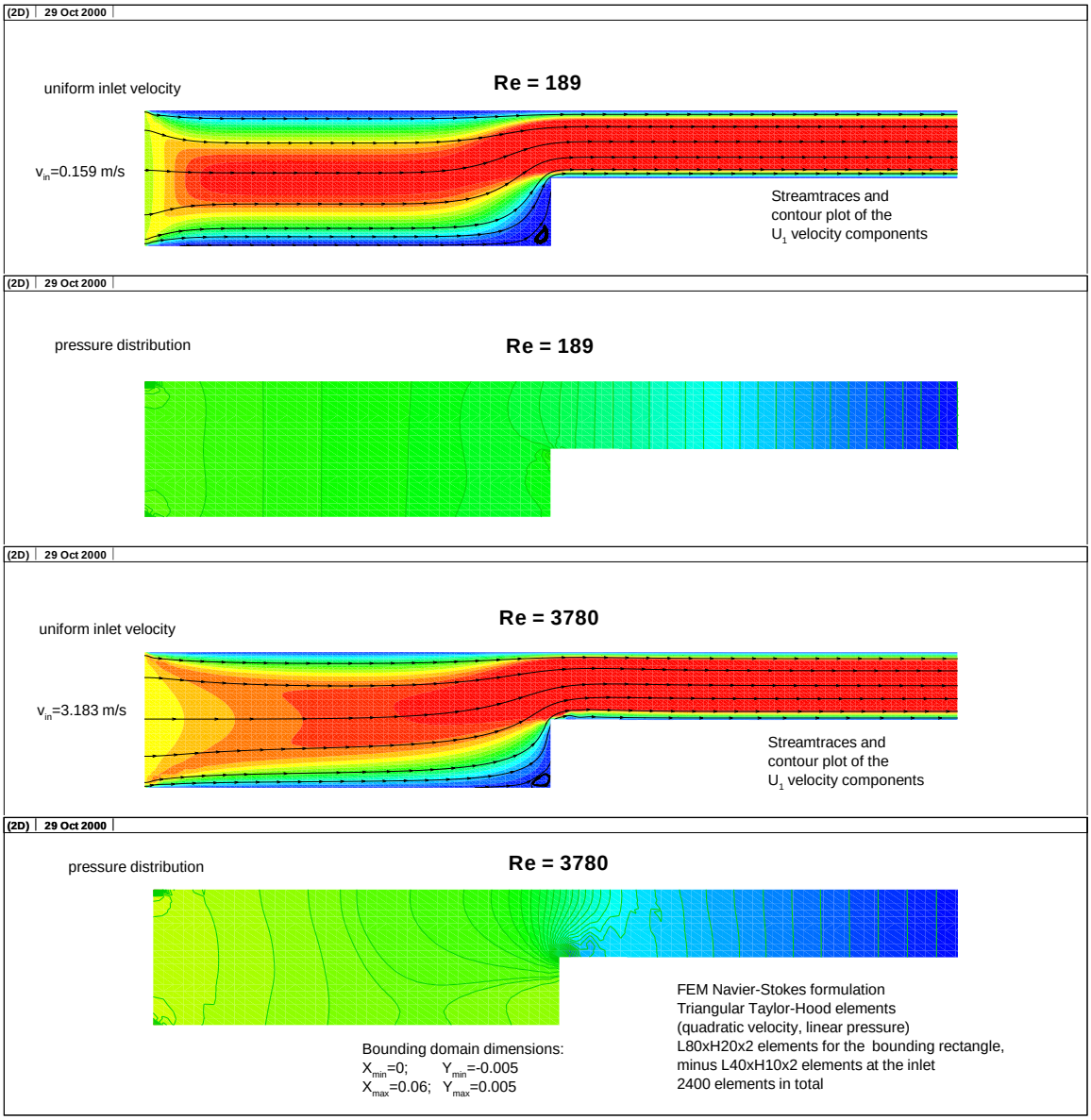


Table N.2: Test matrix for the forward facing step cases

Case#	*Flow rate*	$Velocity_{in}$	Re_{in}	dt_{crit}	dt_{used}	RMS_{crit}	*Iter#*	*CPU*
$ST1a$	0.00005	1	1	n/a	n/a	n/a	n/a	531
$ST1$	0.00005	1	1	n/a	n/a	n/a	n/a	552
$ST2$	0.00005	0.1592	1	n/a	n/a	n/a	n/a	557
$NS1$	0.00005	0.1592	189	0.00057	$14.2E-05$	0.00001	27	615
$NS2$	0.0001	0.3183	378	0.00028	$7.1E-05$	0.00001	75	569
$NS3$	0.0002	0.6366	756	0.00014	$3.5E-05$	0.00001	217	466
$NS4$	0.0004	1.2732	1512	$7.09804E-05$	$1.8E-05$	0.00001	687	679
$NS5$	0.0006	1.9099	2268	$4.73202E-05$	$1.2E-05$	0.00001	1125	1301
$NS6$	0.0008	2.5465	3024	$3.54902E-05$	$8.9E-06$	0.00001	1321	1613
$NS7$	0.001	3.1831	3780	$2.83921E-05$	$7.1E-06$	0.00001	1474	1771
$NS7a$	0.001	3.1831	3780	$2.83921E-05$	$7.1E-06$	0.00001	1286	2029

the capture of a larger re-circulation zone behind the fence. The test conditions are given in Table N.3. Typical examples of the results are shown in Figure N.5, for Re=189 and Re=2268, and Figure N.6 for Re=3780. The flow enters from the left side of the domain with uniform velocity and exits to the right side. (Note that, is is different from the inlet velocity profile shown in the vertical wall sketch.) At low Reynolds-numbers the flow follows the boundaries. There is only a small re-circulation zone on both sides of the domain. With the increase of the Reynolds-number the size of the the downstream re-circulation zone increases. At Re=3780 the re-circulation zone almost reaches the duct exit. In addition, a secondary re-circulation zone can be seen at the top corner of the exit. The robustness of the code is demonstrated by the back-flow, captured by the solution.

Figure N.4: Vertical wall

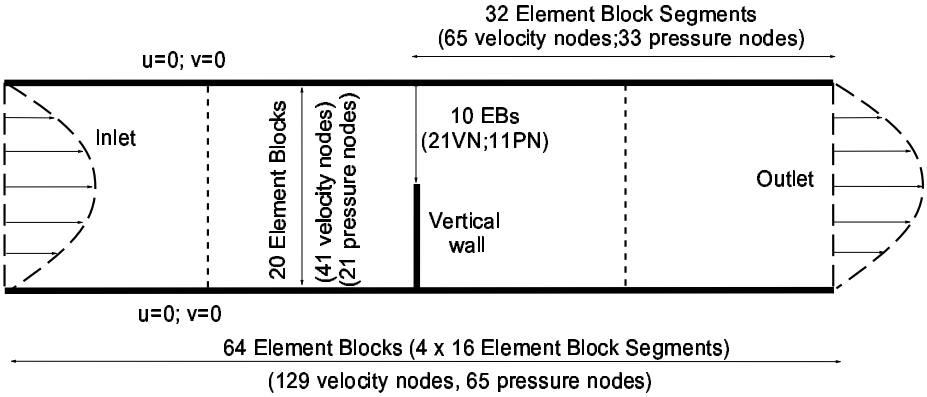


Table N.3: Test matrix for the fence in a duct cases

$Case\#$	$Flow\ rate$	$Velocity_{in}$	Re_{in}	dt_{crit}	dt_{used}	RMS_{crit}	$Iter\#$	CPU
$ST1$	0.00005	1	1	n/a	n/a	n/a	n/a	860
$ST2$	0.00005	0.1592	1	n/a	n/a	n/a	n/a	810
$NS1$	0.00005	0.1592	189	$56.8E-05$	$28.4E-05$	0.00001	19	644
$NS2$	0.0001	0.3183	378	$28.4E-05$	$14.2E-05$	0.00001	53	710
$NS3$	0.0002	0.6366	756	$14.2E-05$	$7.1E-05$	0.00001	190	581
$NS4$	0.0004	1.2732	1512	$7.1E-05$	$3.5E-05$	0.00001	618	885
$NS5$	0.0006	1.9099	2268	$4.7E-05$	$2.4E-05$	0.00001	1276	1268
$NS6$	0.0008	2.5465	3024	$3.5E-05$	$1.2E-05$	0.00001	3190	2990
$NS7$	0.001	3.1831	3780	$2.8E-05$	$9.5E-06$	0.00001	3596	4003

Figure N.5: Sample result for a fence in a duct (1)

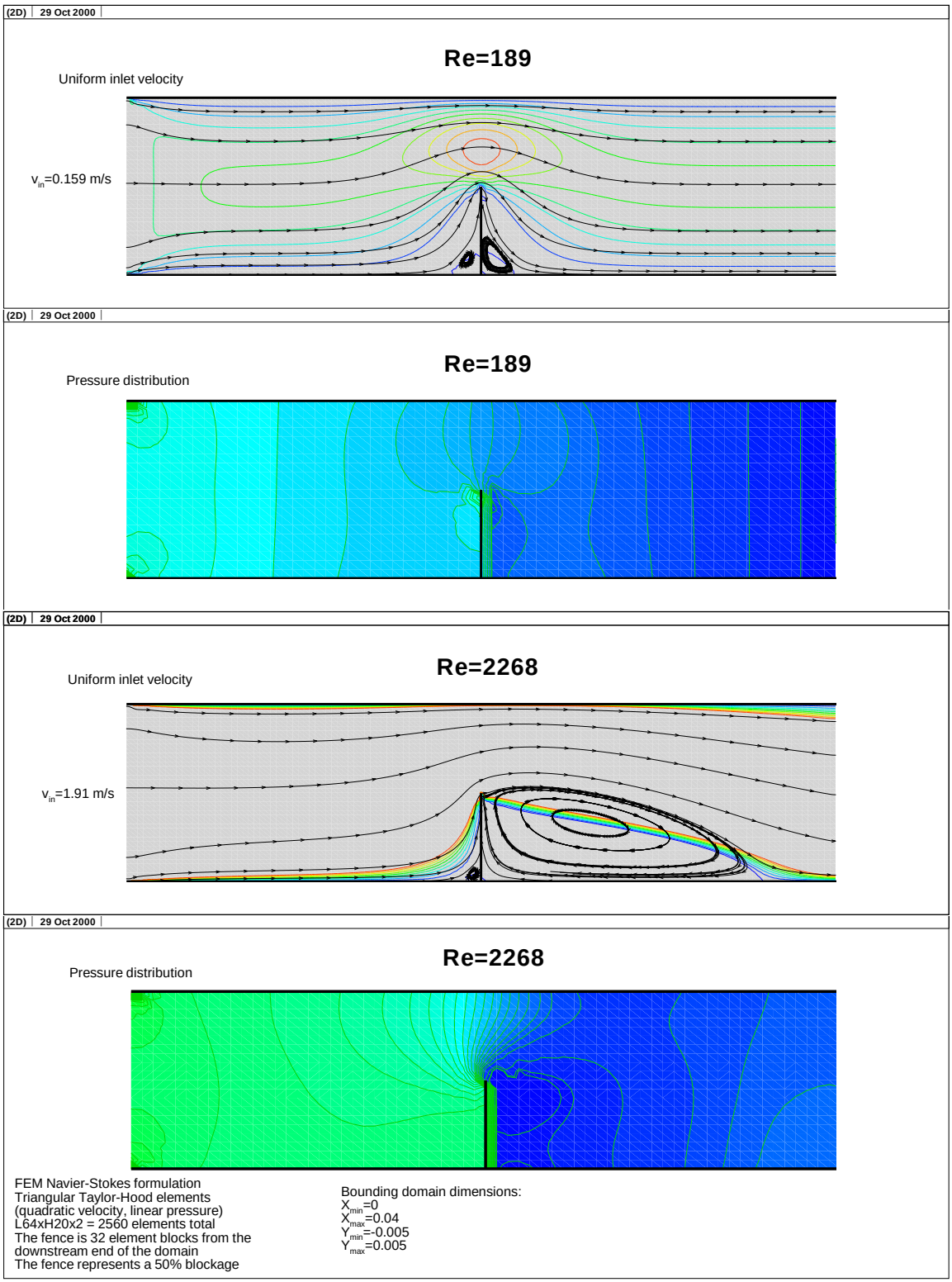
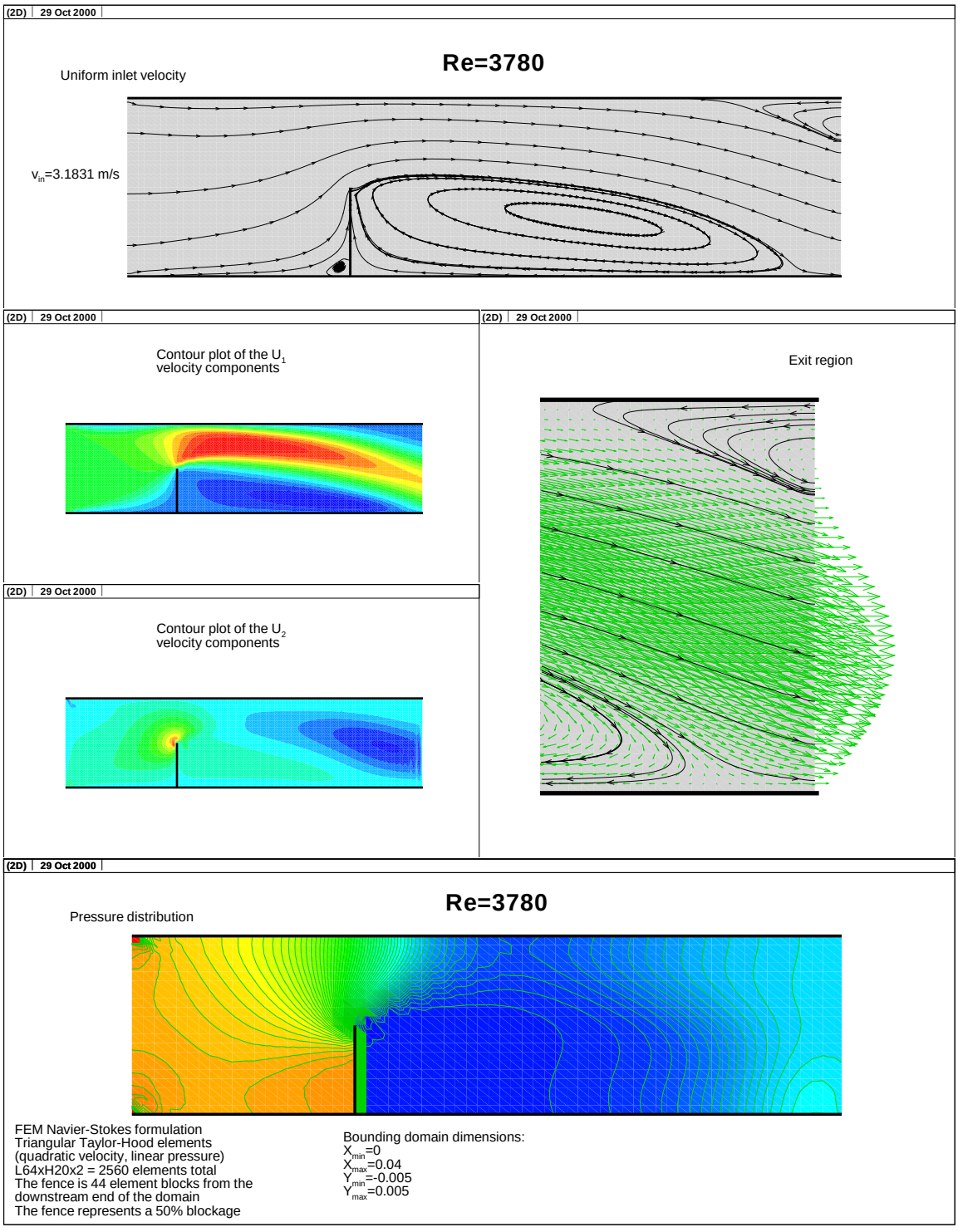


Figure N.6: Sample result for a fence in a duct (2)



Appendix O

Static displacement of the soft palate

This chapter details the results for a two-dimensional duct flow, where the the cantilevered plate that splits the domain at the inlet is statically displaced perpendicular to the streamwise direction. A static displacement of the cantilevered plate was prescribed to the internal domain boundary. The calculations were performed to test the code's capability for predicting pressure trends along the cantilevered plate.

O.1 Methodology and assumptions

Transverse vibration of a rod is described by Nowacki [Now63], providing the eigenfunctions for cantilevered plates, clamped at one and and free at the other end. These equations were used to generate the initial displacements for the first 3 Eigenmodes of the cantilevered plate. The initial nodal displacement was pre-calculated in MS-Excel spreadsheets or later using a FORTRAN code. These pre-calculations allowed for the generation of fundamental mode displacements for any of the examined initial conditions.

Throughout the calculations the plate was discretized to 17 nodes. The output from the main program was formatted in a way, which allowed for easy graphical post-processing with the TecPlot plotting package.

Fluid loading and coupling between the fluid and the wall codes will be considered in the final part of the research. Note that, the discretization of the fluid domain did not account for the plate thickness. However, it will be accounted for in the wall code and can be adjusted through an input variable, as required.

O.2 Results for the static displacement cases

The Navier–Stokes equations for 2D duct flows were programed using an explicit FEM method. Three modifications were implemented to the code. First, the code employs new mesh generators for velocity and pressure nodes, and uses an initial displacement array for the compliant plate section. (Note that at this stage the plate is static throughout the time series. The time series is stopped when a quasi-steady-state conditions is reached.) Second, the pressure is dimensionalized in the code using the ρU_{in}^2 scaling. The pressure distribution is given in kPa. Third, the pressure forces are saved for each node of the compliant plate. This will be used in the wall code as explained earlier.

Two sets of runs were conducted. The test matrix with the run ID's are given in Table O.1 The Stokes and Navier-Stokes cases were performed using $Re = 1$ and $Re = 1512$ and $U_{in} = 1$ and $U_{in} = 1.273$, respectively. For the Navier-Stokes cases the time step was set to $\Delta t = 5 \times 10^{-5}$ and for convergence limit $RMS = 1 \times 10^{-5}$ was used.

The domain was discretized to 80 element blocks along the x-axis and 16 element blocks along the y-axis with two triangular elements in each rectangular element block, resulting in a 2560 element domain. The corresponding physical dimensions were $L = 0.04$ and $h = 0.01$. The inlet was split in the middle, 8 element blocks from the bottom left corner, 30 element blocks along the x-axis. The last 15 element blocks of the mid-line split were designated to the com-

Table O.1: Test matrix for static displacement cases

Case type	Max. Displ.	Eigenmode 1	Eigenmode 2	Eigenmode 3
Stokes	0.0005	$StM1W0.0005$	$StM2W0.0005$	$StM3W0.0005$
Stokes	0.001	$StM1W0.001$	$StM2W0.001$	$StM3W0.001$
Stokes	0.002	$StM1W0.002$	$StM2W0.002$	$StM3W0.002$
Navier – Stokes	0.0005	$NsM1W0.0005$	$NsM2W0.0005$	$NsM3W0.0005$
Navier – Stokes	0.001	$NsM1W0.001$	$NsM2W0.001$	$NsM3W0.001$
Navier – Stokes	0.002	$NsM1W0.002$	$NsM2W0.002$	$NsM3W0.002$

pliant section. This resulted in 16 + 1 pressure nodes, where the additional nodes were used for the non-moving dummy nodes, due to the clamped boundary condition.

The results are shown in Figures O.1, O.2 and O.3 for the $w = 0.0005$, $w = 0.001$ and $w = 0.002$ maximum displacement cases, representing 10%, 20% and 40% obstructions in the inlet channels. The top, middle and bottom figures show the x and y velocity and pressure distributions, respectively.

Figures O.4, O.5 and O.6 give the pressure distribution along the displaced "soft palate" for the Mode 1, Mode 2 and Mode 3 test cases. The top figures show the initial displacements, while the middle and bottom figures give the corresponding pressure distributions for the Stokes and Navier-Stokes cases. The corresponding normalized results are shown in Figures O.7, O.8 and O.9, respectively.

Figure O.1: Mode 1 static displaced "soft palate", v-p distribution

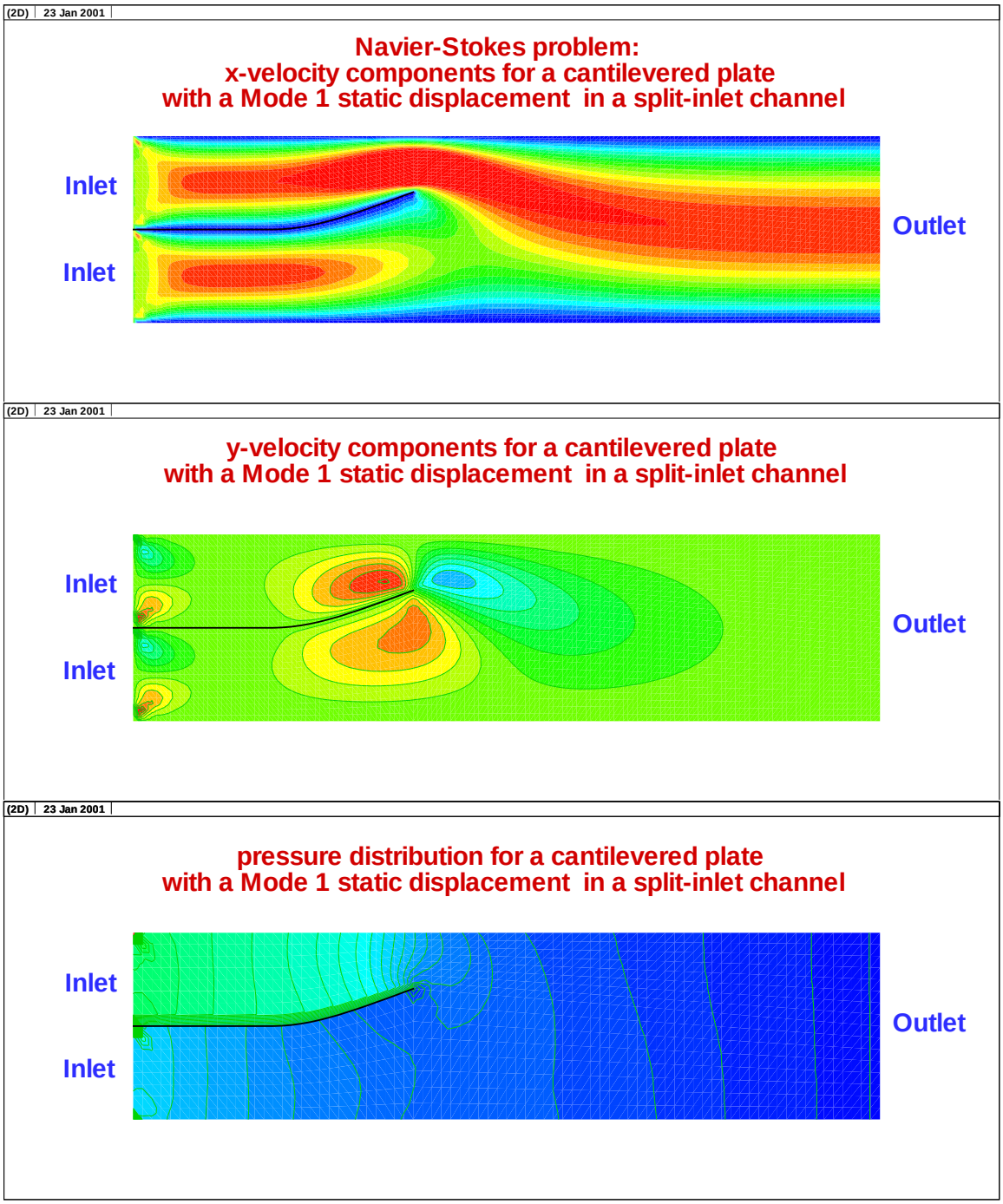


Figure O.2: Mode 2 static displaced "soft palate", v-p distribution

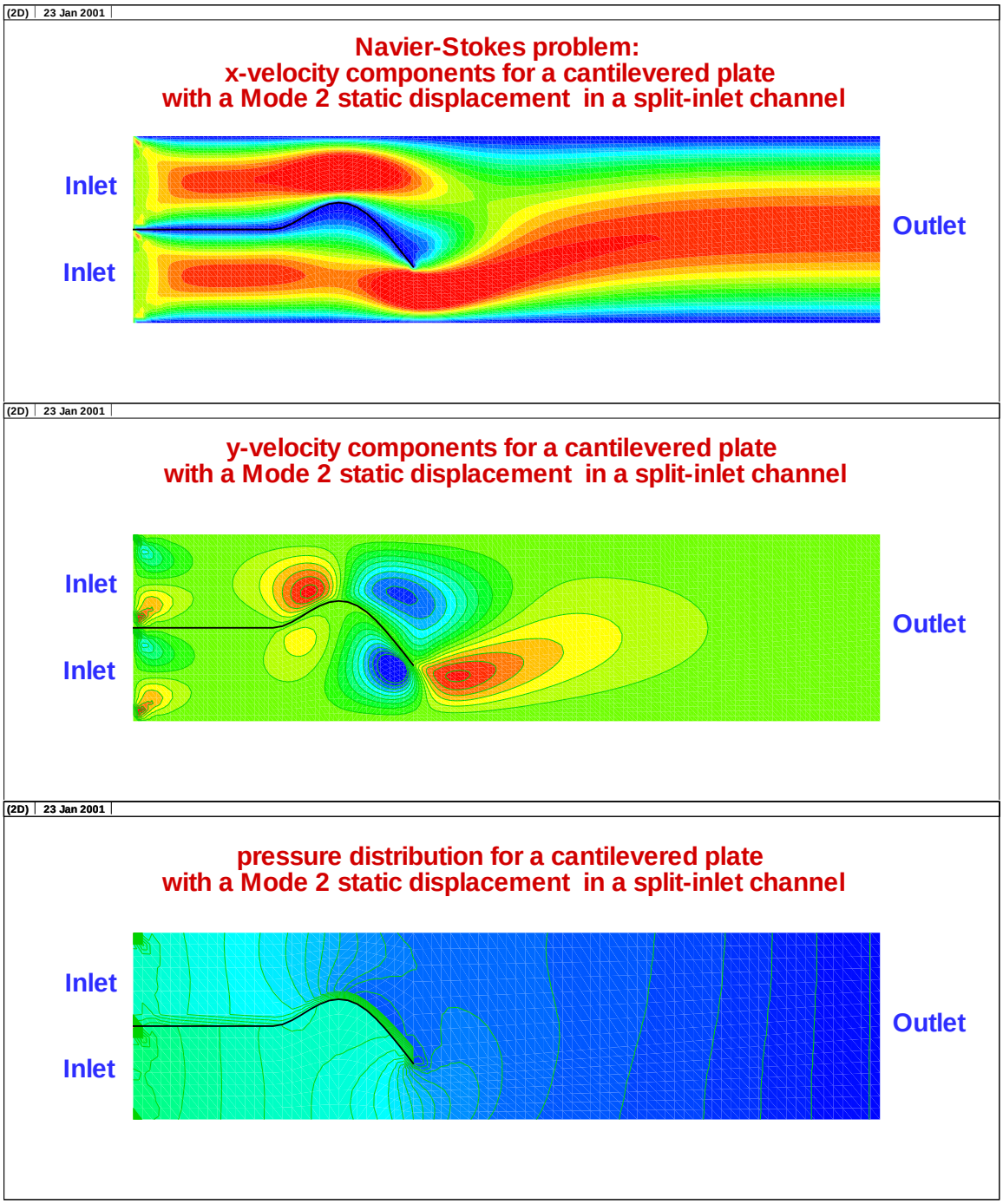


Figure O.3: Mode 3 static displaced "soft palate", v-p distribution

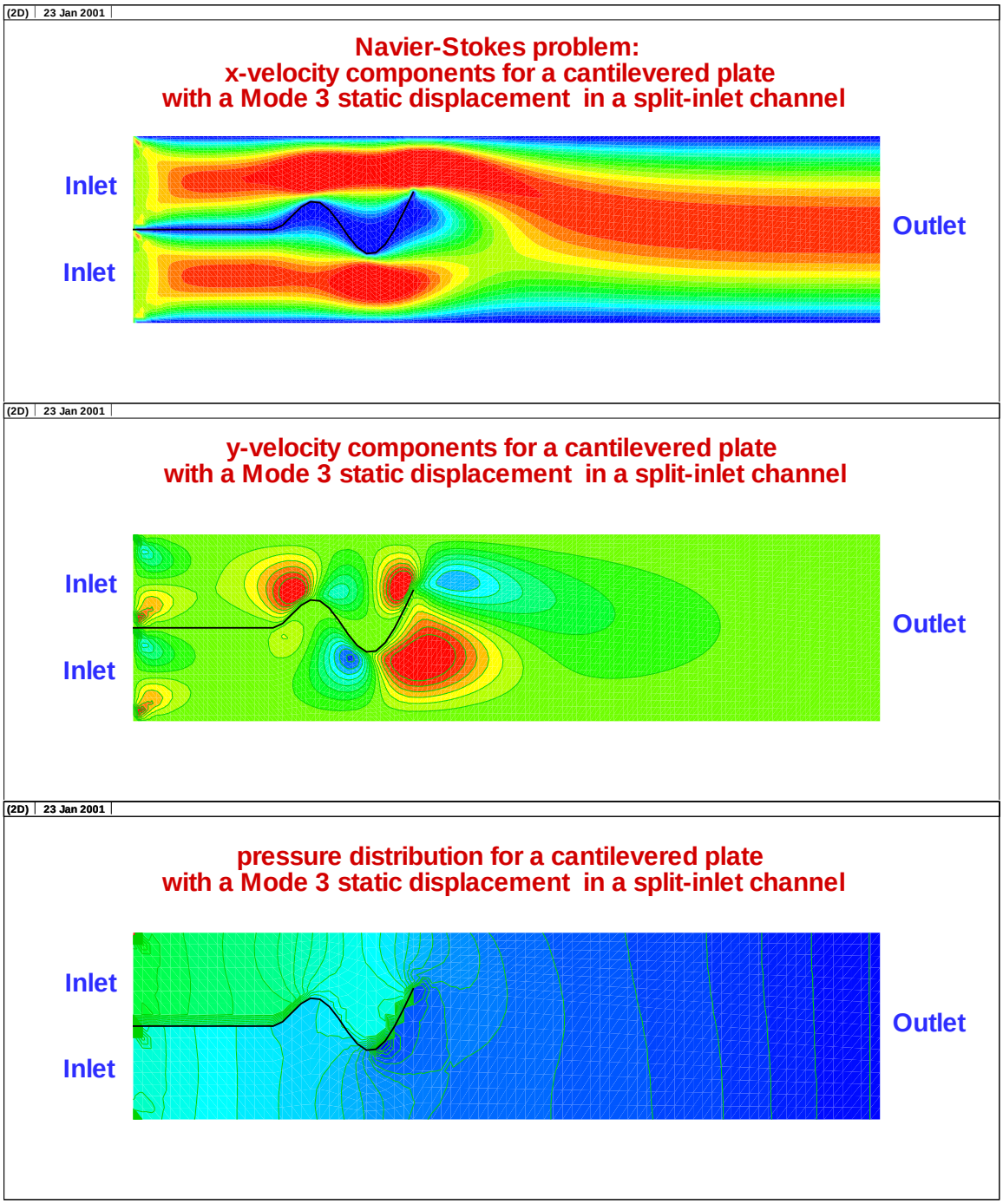


Figure O.4: Pressure profile around a Mode 1 displaced cantilevered plate

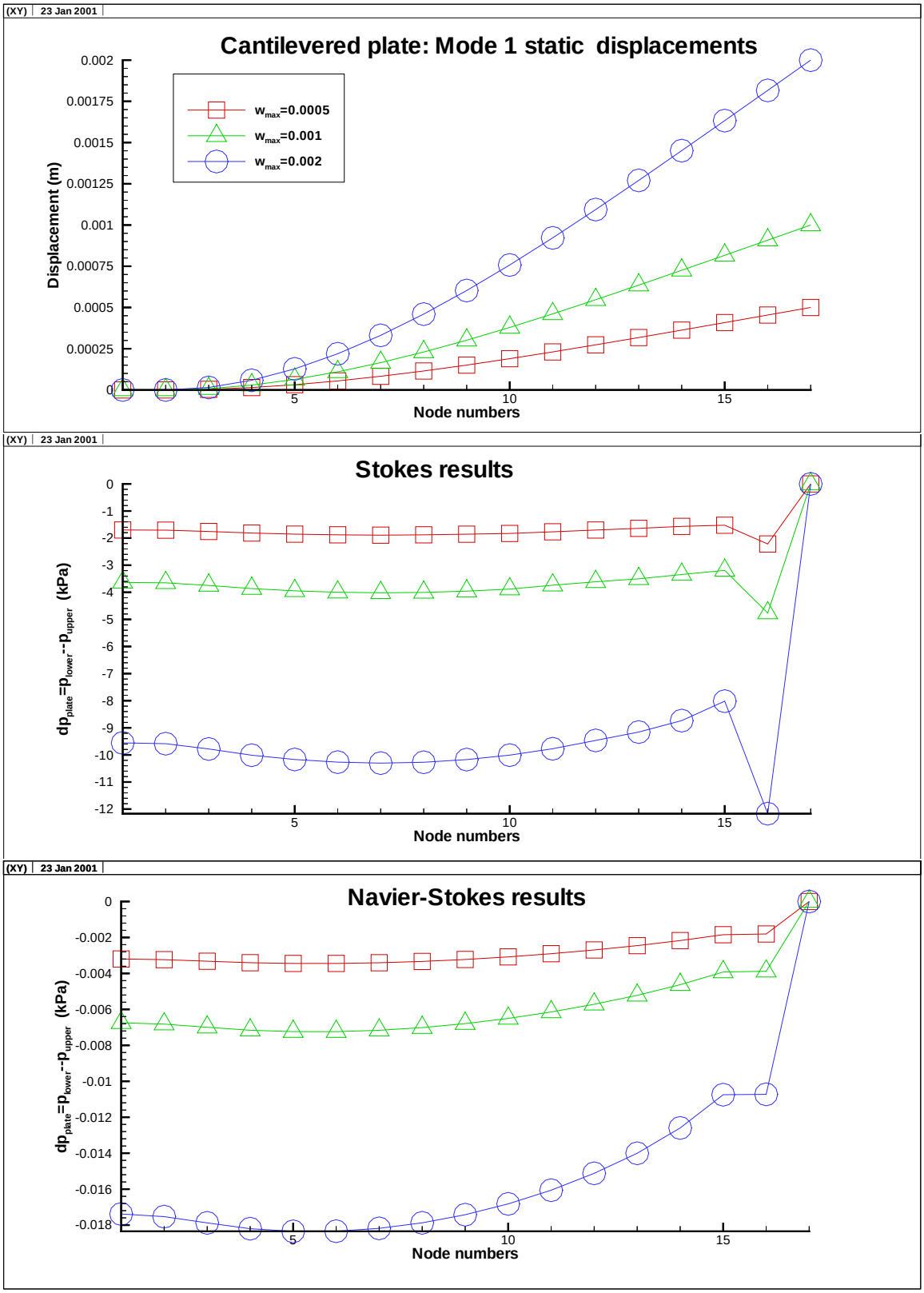


Figure O.5: Pressure profile around a Mode 2 displaced cantilevered plate

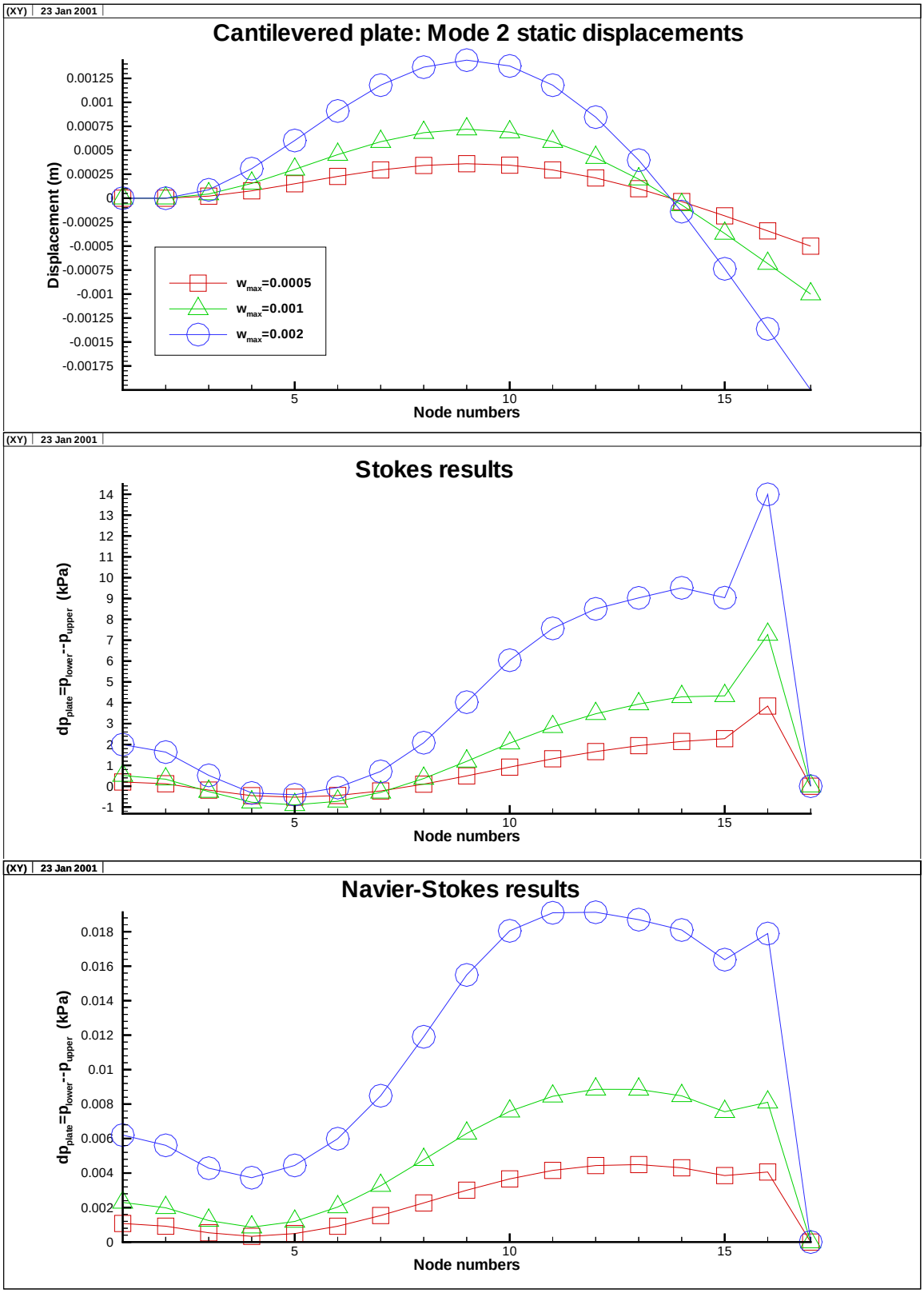


Figure O.6: Pressure profile around a Mode 3 displaced cantilevered plate

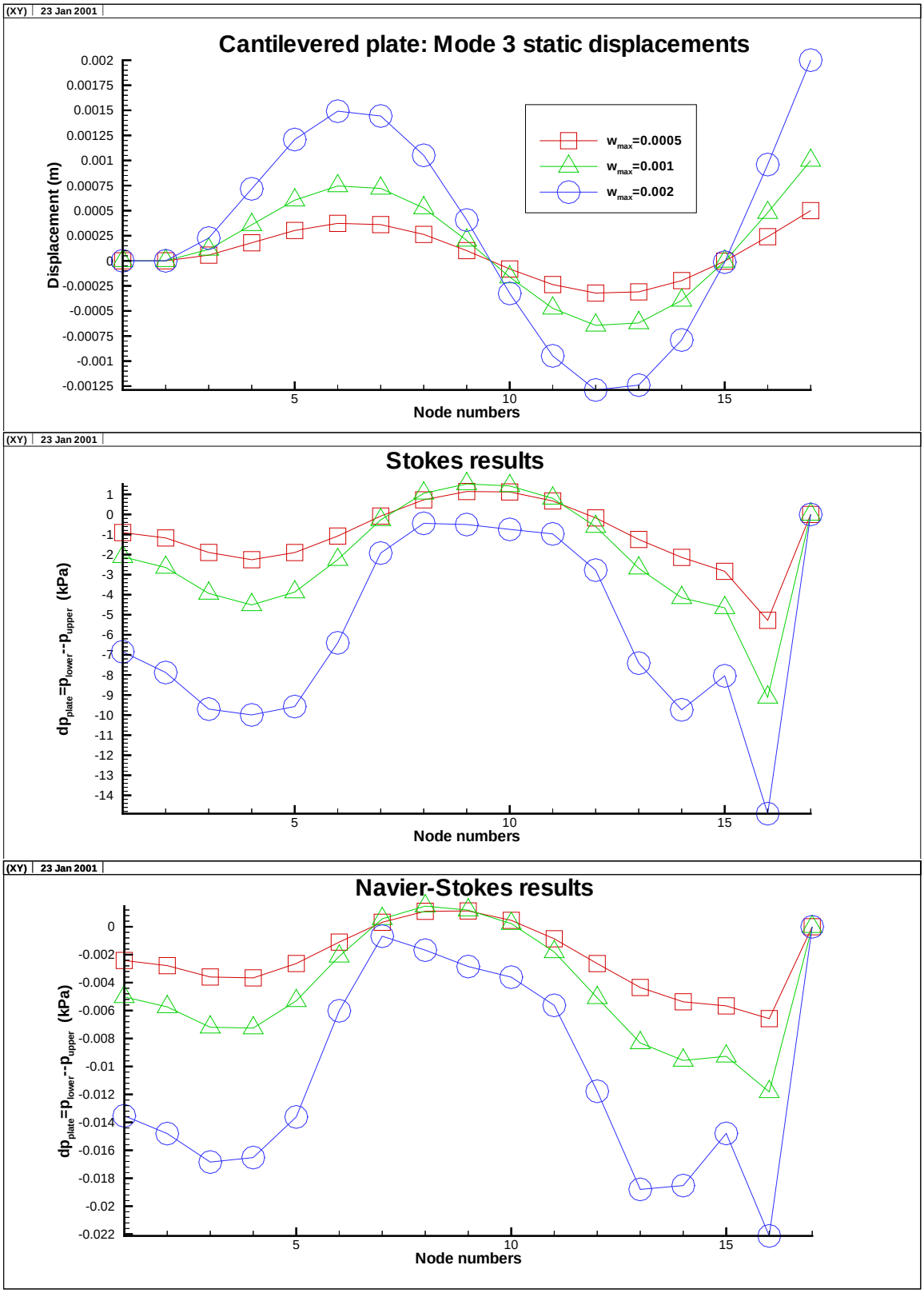


Figure O.7: Normalized pressure profile around a Mode 1 displaced cantilevered plate

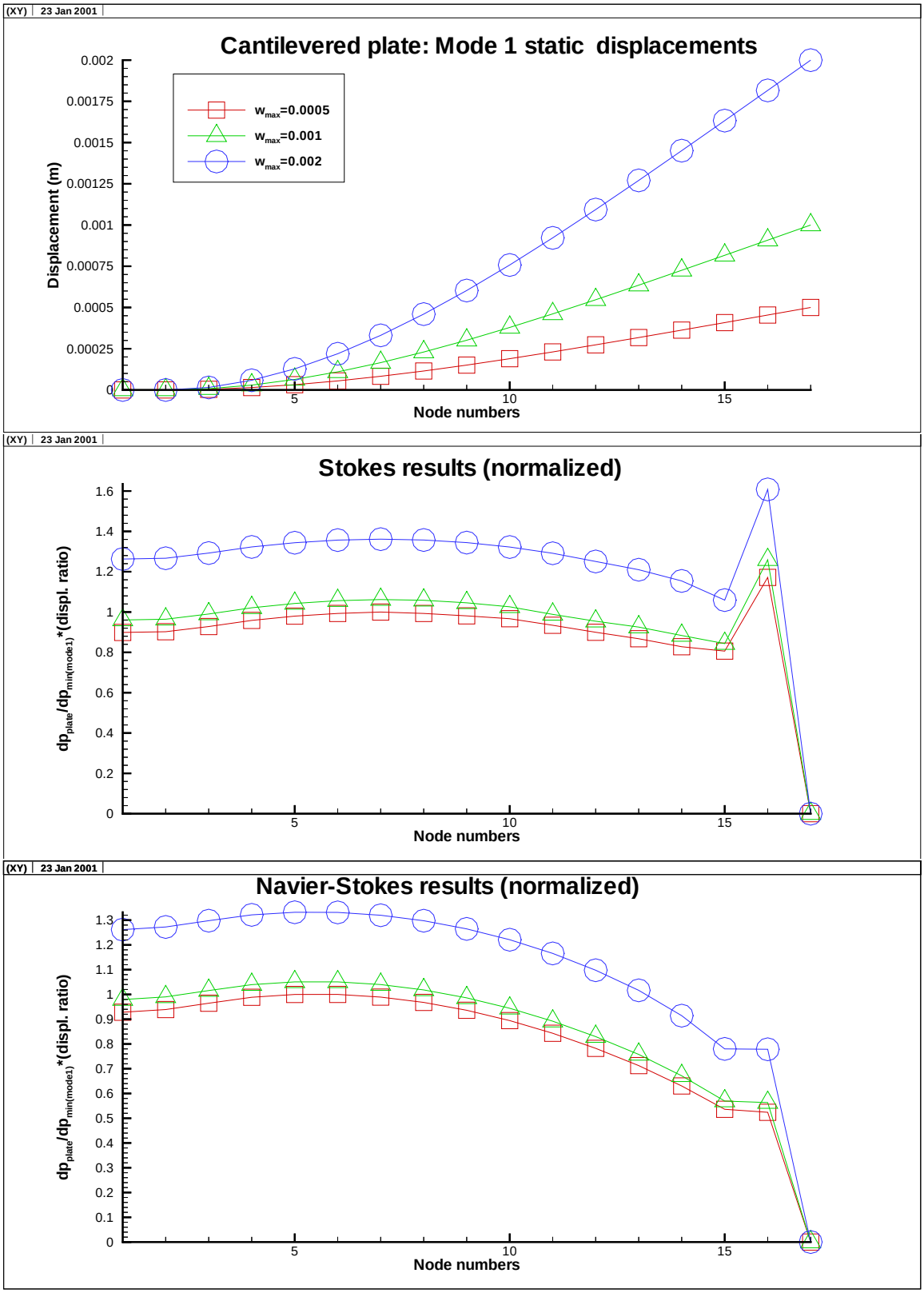


Figure O.8: Normalized pressure profile around a Mode 2 displaced cantilevered plate

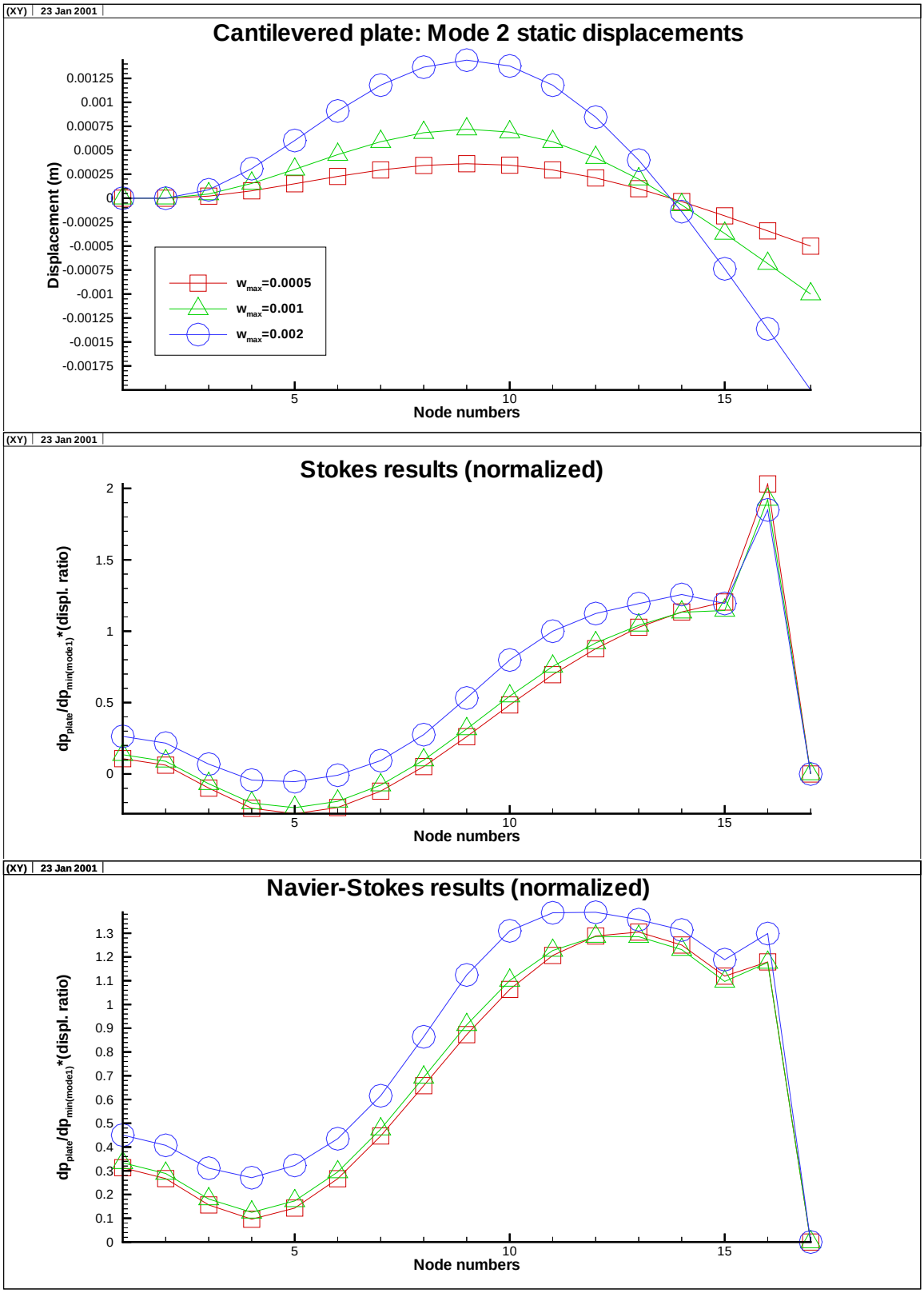
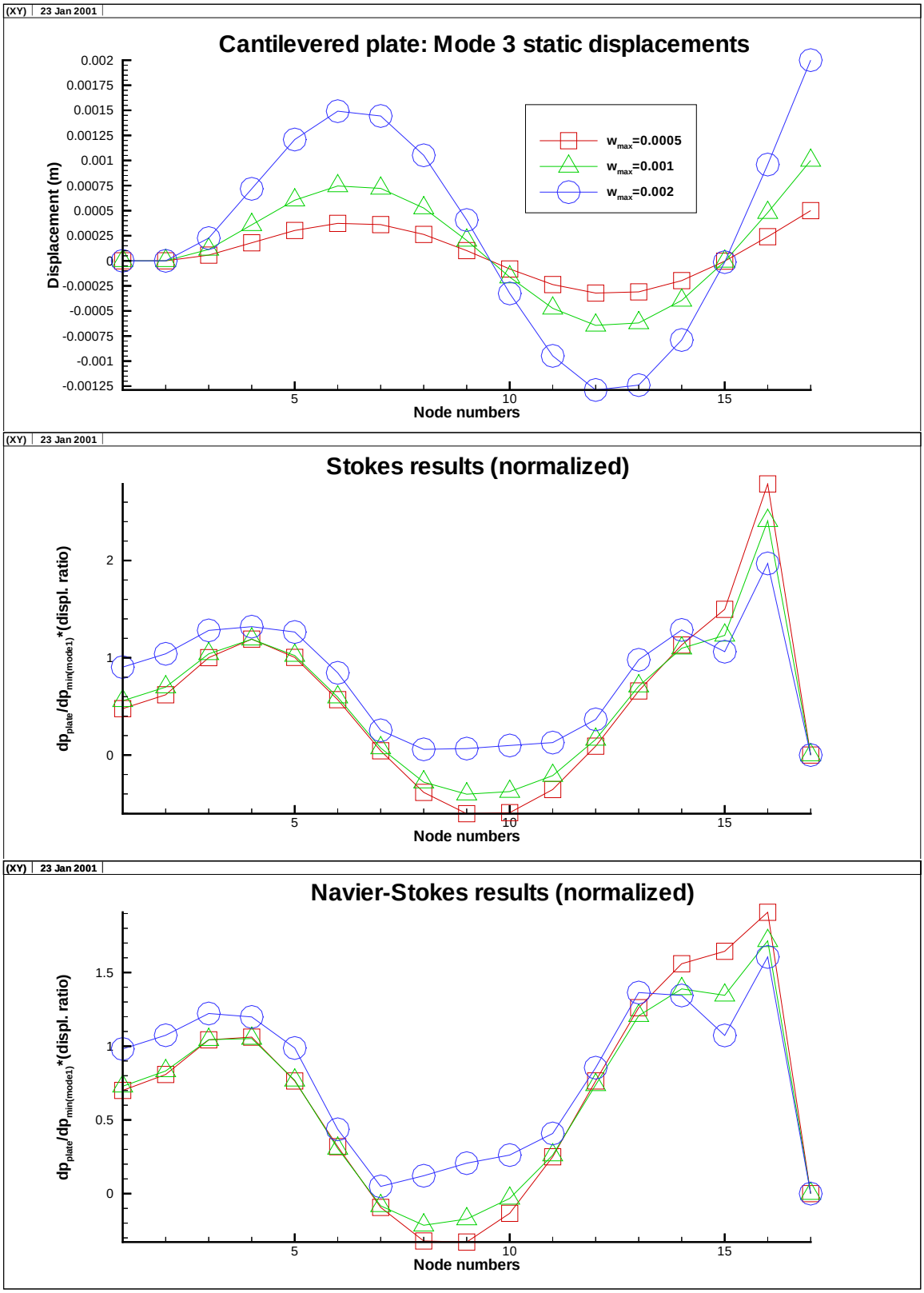


Figure O.9: Normalized pressure profile around a Mode 3 displaced cantilevered plate



Appendix P

Internal flow examples from open literature

Figure P.1: Lid-driven cavity flow, Re=0.017, corner eddies in a creeping flow [Dyk82]

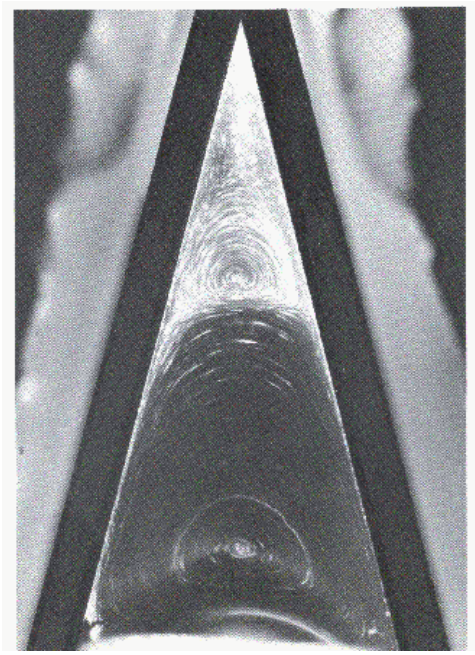


Figure P.2: Shear layer driven cavity flow, square domain, Re=0.01 [Dyk82]

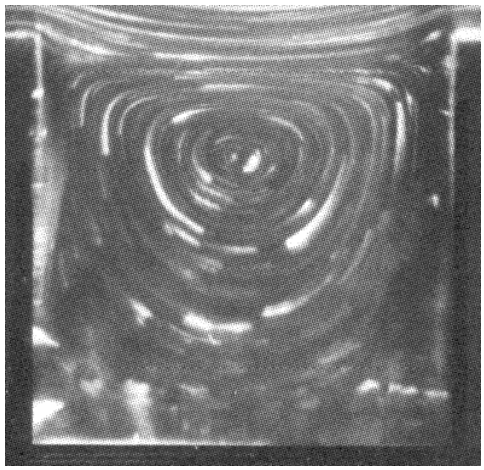


Figure P.3: Lid-driven cavity flow, square domain, Re=0.017 [Tei97]

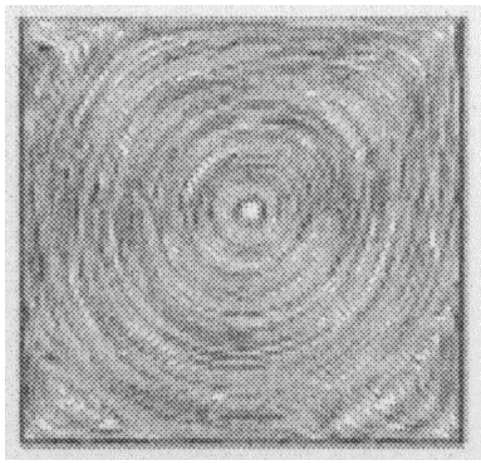


Figure P.4: Numerical results for lid-driven cavity flow, square domain [Tei97]

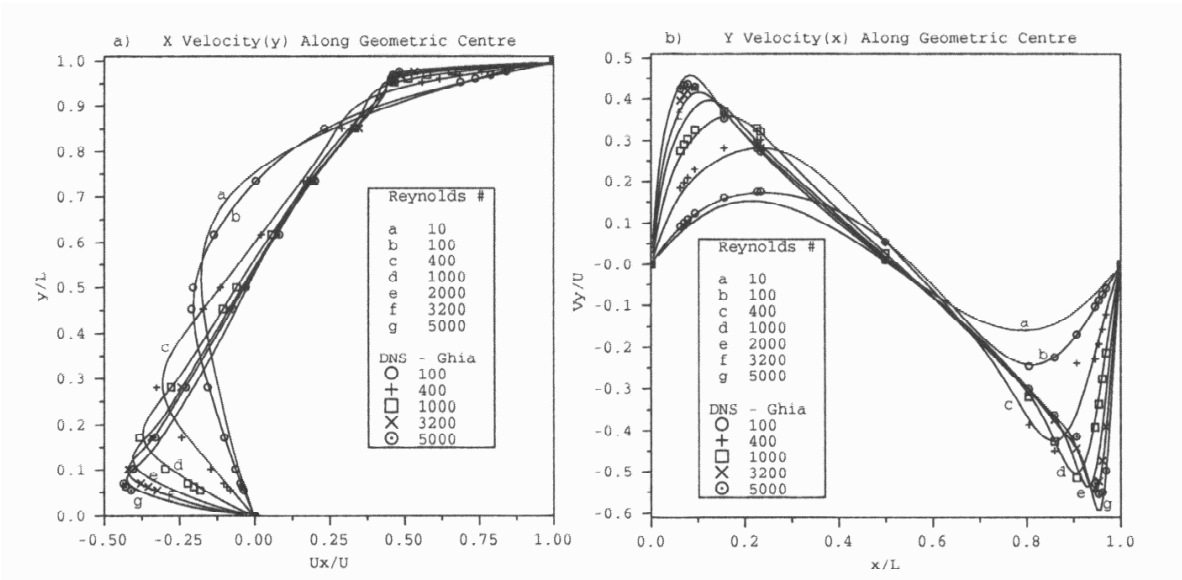


Figure P.5: Steady flow in a driven cavity. Streamline patterns (a) Re=400; (b) Re=1000; (c) Re=4000; (d) Re=10000 [She90]

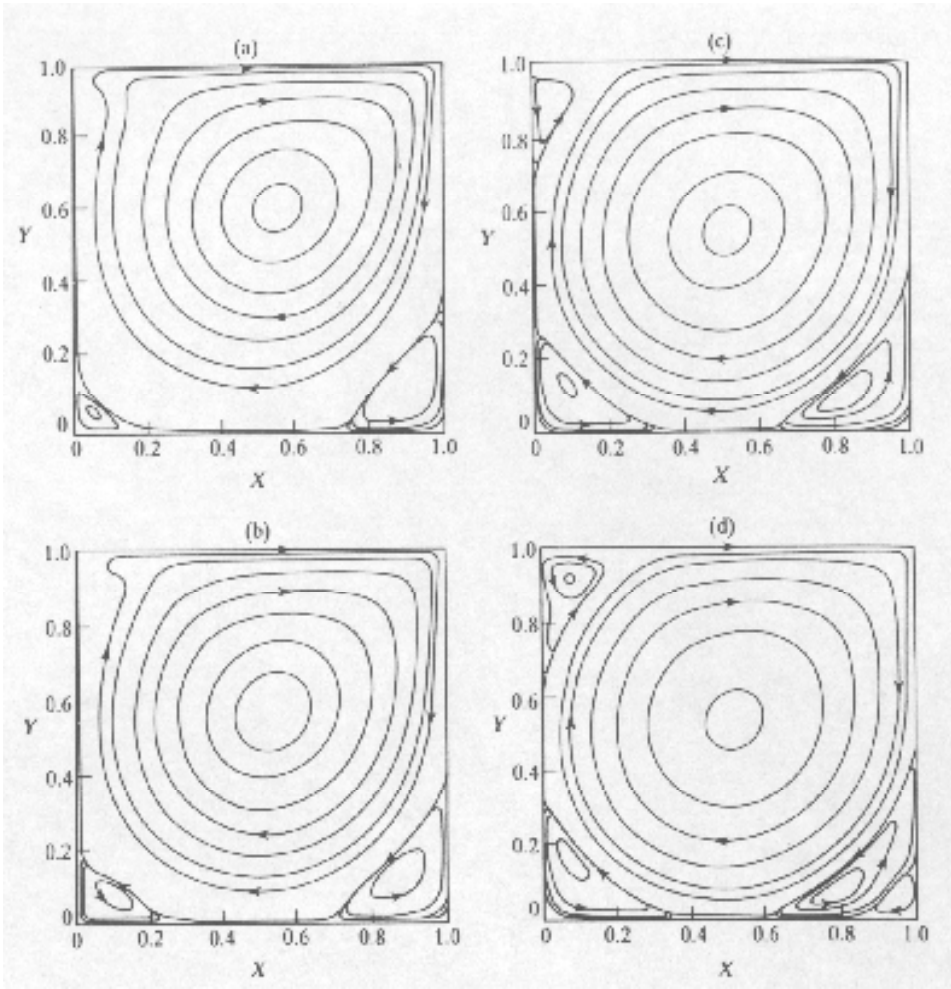


Figure P.6: Fence in a channel, Re=0.014 [Dyk82]